Re-Sonification of Objects, Events, and Environments

by

Alex M. Fink

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2013 by the
Graduate Supervisory Committee:

Andreas Spanias, Chair
Pavan Turaga
Perry Cook
Konstantinos Tsakalis

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Digital sound synthesis allows the creation of a great variety of sounds. Focusing on interesting or ecologically valid sounds for music, simulation, aesthetics, or other purposes limits the otherwise vast digital audio palette. Tools for creating such sounds vary from arbitrary methods of altering recordings to precise simulations of vibrating objects. In this work, methods of sound synthesis by re-sonification are considered. Re-sonification, herein, refers to the general process of analyzing, possibly transforming, and resynthesizing or reusing recorded sounds in meaningful ways, to convey information. Applied to soundscapes, re-sonification is presented as a means of conveying activity within an environment. Applied to the sounds of objects, this work examines modeling the perception of objects as well as their physical properties and the ability to simulate interactive events with such objects.

To create soundscapes to re-sonify geographic environments, a method of automated soundscape design is presented. Using recorded sounds that are classified based on acoustic, social, semantic, and geographic information, this method produces stochastically generated soundscapes to re-sonify selected geographic areas. Drawing on prior knowledge, local sounds and those deemed similar comprise a locale's soundscape.

In the context of re-sonifying events, this work examines processes for modeling and estimating the excitations of sounding objects. These include plucking, striking, rubbing, and any interaction that imparts energy into a system, affecting the resultant sound. A method of estimating a linear system's input, constrained to a signal-subspace, is presented and applied toward improving the estimation of percussive excitations for re-sonification.

To work toward robust recording-based modeling and re-sonification of objects, new implementations of banded waveguide (BWG) models are proposed for object modeling and sound synthesis. Previous implementations of BWGs use arbitrary model pa-

rameters and may produce a range of simulations that do not match digital waveguide or modal models of the same design. Subject to linear excitations, some models proposed here behave identically to other equivalently designed physical models. Under nonlinear interactions, such as bowing, many of the proposed implementations exhibit improvements in the attack characteristics of synthesized sounds.

DEDICATION

Dedicated to unreasonable men, readers and travelers.

ACKNOWLEDGEMENTS

This work has been made possible by numerous people who have given me help along the way. Somebody else made this happen, so to speak. My education certainly would not have been possible without the support of my family, financial and otherwise.

Spending these years in Tempe, or anywhere, would likely not have been tolerable if not for the presence of great friends, some of whom have passed on to better places.

A special thanks goes to my colleagues and coworkers in Arts, Media and Engineering and the SenSIP lab for great support, and for always being available to answer questions on a variety of topics.

Much gratitude needs to be directed to the various staff at ASU who tolerated my loathsome attitude toward paperwork and procedure, as well as a slew of asinine questions.

Thank you to the National Science Foundation, for funding my IGERT fellowship with the (now) School of Arts, Media and Engineering. Also, thanks to the School of Electrical, Computer and Energy Engineering for providing funding in the form of a teaching assistantship, and thanks to Prof. Kozicki, Prof. Aberle, Mayda, and others for being great to work with.

I would like to thank my committee and others who have provided me with academic guidance and taken the time to provide me with feedback. Thanks to Prof. Andreas Spanias for providing administrative and academic support throughout my tenure at ASU, and for taking me on as a student in my final few years. Thanks to Prof. Perry R. Cook, who has provided expert guidance and wonderful feedback as a committee member, taking the time for many trips and teleconferences. Thanks to Prof. Pavan Turaga for reaching out to me and serving on my committee. Thanks to Prof. Konstantinos Tsakalis for constantly making himself available and serving on my committee.

This work has also been made possible through the guidance of Dr. Harvey Thornburg. A special thanks to him for handling many drafts of many papers, providing navigation through literature and ideas, and fostering an appreciation of information.

This material is based upon work supported by the National Science Foundation under Grant No. 0504647. This work is also supported in part by the NSF Net Centric I/UCRC - ASU SenSIP site Award 1035086.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

This document describes efforts to improve sound creation, editing, and understanding. This stems from an assumption that things related to sound, like much of science, are hard, and from a desire to work toward usable and useful tools for making sounds. This is primarily addressed with two different approaches: creating combinations of sounds (i.e., environments or soundscapes) and creating individual sounds (i.e., the isolated sounds of objects and events). The methods herein are focused on using sound recordings as a starting point from which new sounds are made, though the described object/event models may be created from physical, perceptual, spectral, or temporal parameters.

For environmental re-sonification, a method for automated soundscape creation for designing explorable sound maps is presented. This method serves as a sort of complementary synthesis end to the work of Dr. Gordon Wichern in segmenting, indexing, and retrieving environmental and natural sounds, and also relies on work by colleagues and collaborators including Dr. Harvey Thornburg, Brandon Mechtley, Jiachen Xue, and Jinru Liu. Extensions and continuation of environmental re-sonification may be found in the work of Mr. Mechtley.

The larger focus here is on source/filter or exciter/resonator approaches to describing isolated sounds, primarily of simulated resonant objects. Such schema attempt to meaningfully and separately represent objects and ways to interact with them. To this end, a method of estimating percussive excitations is presented, as are suggested alterations to banded waveguide models. Banded waveguide models were chosen as a focus of object modeling given their status as a hybrid spectral/physical model, meaningfully representing physical and perceptual properties of modeled objects. The cause for studying differing implementations of banded waveguide models stems from the author's observation of their variant behavior and difference in simulated outputs, especially in

comparison to oft studied and implemented digital waveguide models. In improving the correspondence of banded waveguide models with other models for sound synthesis, applications may make use of the many methods for re-sonification for these other models.

Chapter 1

INTRODUCTION

In this work, methods of re-sonifying objects, events, and environments are presented. This refers to meaningfully reusing recorded sounds in some way, often analyzing, possibly transforming, and resynthesizing them. Re-sonification may be used in applications ranging from musical composition to interaction in virtual environments, where sounds need to be easily and meaningfully manipulable. Considering sound as a carrier of information about activity, meaningful re-sonification refers to methods that preserve ecological validity, giving listeners a sense of real events that occur between real objects within real environments, or methods that achieve any desired sonic effects.

Working toward scalable and robust re-sonification, this work addresses re-sonification at two scales: the object/event level and the environmental level. For object re-sonification, banded waveguides are examined for modeling of resonant sounding objects. Study of event re-sonification herein is focused on estimating percussive excitations for resynthesis. A multiply-informed methodology for selecting and playing back source recordings is explored for re-sonifying environments.

As with any work, much is built upon the efforts of others. Where appropriate, such sources are cited, though a few stand out in significance and number of citations. With much focus on physical models, this document uses [2] as a significant source of background material. Additionally, information about and justification for use of banded waveguide models draws largely from [6,7]. Many original contributions described here have appeared in [8–12].

1.1 Motivation

Re-sonification is employed in a variety of applications where sounds need to be meaningfully modified or used. For example, hearing aids benefit from noise suppression

and processing to enhance speech perceptibility [13]. In musical applications, film creation, and virtual interactions, re-sonification of sounds allows for the creation of both realistic and fantastic sounds [14]. For soundscape creation, sounds must be created or used to meaningfully represent the activity of an environment [15]. To accomplish re-sonification in such varying applications, tools must be on-hand to analyze, understand, model, recreate, reuse, or alter sounds in desired fashions.

1.2 Overview of Re-Sonification of Environments

At the scale of environments, we consider re-sonification as a means of using existing sound recordings to create soundscapes (i.e., sonic textures made of sounds individually classifiable in some sense as those of objects and events) of virtual environments to give a sense of place. For example, to re-sonify the sound of a busy intersection, we wish to play a mix of sounds recorded there, as well as any other relevant sounds. Using the sounds from other intersections, such as those of cars and audible crosswalks, might help to enrich the soundscape, but choosing appropriate sounds and how to use them is not a trivial task.

Much work in the re-sonification of environments focuses largely on how to design soundscapes. Typically relying on the sequencing of shorter source recordings, soundscape design requires informed selection of component sounds to provide meaningful sonic immersion. Many approaches rely on careful design via a "composer" [16] relying on his/her prior knowledge or that gained through means such as interviews [17]. We seek ways, however, to automatically design soundscapes for environmental re-sonification.

This work therefore presents a method for the automated design and implementation of interactive soundscapes for exploring activity within geographic regions. This method, introduced in [12], relies on the use of geo-tagged sound recordings us-

ing the ontological framework of [18], where sounds are related by acoustic features and community-provided semantic tags. By automating soundscape design through the use of geographic, acoustic, semantic, and social information, a scalable process for re-sonifying geographic activity is made possible.

1.3 Overview of Re-Sonification of Objects and Events

At the scale of objects and events, we consider re-sonifying "individual sounds" of the interactive events involving sounding objects. For example, from the recording of a sheet of metal being struck by a screwdriver, we wish to model the metal sheet and the striking of it by the screwdriver. The representation of the exciting action of the screwdriver could be used to virtually strike other objects or a virtual version of the metal sheet. Modeling the metal sheet allows one to change various properties, changing resulting sounds and allowing for alteration of perceived physical characteristics, such as size and material.

Previous work in the re-sonification of objects and events is largely based on spectral [19] and physical [2] models. Spectral models decompose sounds into distinct frequency components, usually varying in time. As a coarse approximation of how humans hear sounds, they are well-suited for re-sonification, with many approaches to spectral analysis established. Spectral models, however, can provide difficulties in providing realistic alterations of interactive events [6]. Alternatively, physical models simulate the vibration of sounding objects to varying degrees of accuracy. Physical models can provide realistic simulations of objects and events, but they are generally difficult to design and control. Additionally, some physical models have a very high complexity in implementation [2].

Working toward broad re-sonification of objects and events, we examine and further develop banded waveguide models, a hybrid approach of physical models and

spectral models [6], well-suited for modeling resonant sounding objects. Use of banded waveguides for re-sonification [20, 21]may rely on existing modal analysis methods [22] to construct models, possibly with some prior knowledge about objects [6]. However, current implementations do not produce results that strongly correspond with comparable modal models [3], particularly with regards to decay rate. Further, as compared to conventional physical modeling methods, banded waveguides may not preserve the transient nature of many sounds [8]. New topologies [8, 9] are therefore presented here for implementing banded waveguides, and improvements are shown in their correspondence with existing, comparable modal and digital waveguide models.

Strictly defining events in a general sense is tenuous [23], but as sound is a concern here, we are interested in events through which sound is manifest. That is, a transfer of energy that creates sound is herein considered an event. With re-sonification as a goal, useful and meaningful representations of events are considered, barring excessive pedantry. To improve the re-sonification of events (specifically, interactions with modeled objects), this work presents the introduction of a duration constraint in estimating percussive sound excitations [10, 11]. The constraint explicitly incorporates prior knowledge about the time-limited nature of percussive interactions in the estimation procedure. We demonstrate improvements using this constrained estimation procedure for re-sonification, as compared to unconstrained methods, especially in the presence of noise.

1.4 Nomenclature, Notation, and Notes

This work includes both esoteric (and possibly new) terminology and non-standard use of existing terms. This brief section serves to introduce and clarify such terminology, in addition to providing clarification of notation and other aspects of this work. The writing of this document has been such that the reader is assumed to have

a basic familiarity with digital signal processing, differential equations, and Newtonian mechanics.

The term *sonification* commonly refers to creating sound that represents information in some way. Defined by [24], sonification is "the technique of rendering sound in response to data and interaction"; similar definitions are offered by others [14,25]. As this work considers the creation of sounds ranging from that of struck resonant objects to cacophonous soundscapes, "sonification" will be used as a catch-all term for the production of digital sounds that convey information about objects, events, or environments. Further, with sound recordings as the source of parameters in the various synthesis methods described, the term "re-sonification" sees significant use. Other work [26–29] employs the term, "resynthesis," but this is generally used with regards to isolated sounds.

Though emphasizing digital signal processing methods for sound synthesis, this work makes use of many terms from other fields such as acoustic ecology, cognitive psychology, and philosophy. With the synthesis of sounds as the objective, usage of such expressions is strongly tied to the context. For example, if we were to consider a recording of a percussionist jumping up and down, then striking a bar with a mallet, the *event* of interest may vary. If physically-based models are being used to recreate or modify the sound of the struck bar, then the imparting of energy from the mallet to the bar is considered the event. When re-sonifying environments via soundscapes, the various audible aspects of the percussionist's performance, such as his/her jumping, are all integral to the event of the bar being played. The philosophical scrutiny of what an event is [23], as with other terms and phrases, should not be critical in reading this work.

In examining the spectra of signals and systems, one may refer to *resonant* frequencies and *modal* frequencies. Though these are related, they are generally not equivalent [30]. Unless noted elsewhere, a *resonance* refers to a spectral peak in a DTFT that

occurs at a *resonant* frequency. *Modal* frequencies describe pole angles in digital models, or equivalent modes of vibration as described in [31]. Resonance and poles in digital models are addressed in detail in Appendix A.

For any object or signal that exhibits multiple resonant or modal frequencies, these peaks or poles will be referred to as *partials*. The lowest in frequency (excepting zero/DC frequency) may be referred to as the *fundamental*, occasionally dubiously so when sounds are not periodic or *pitched* [32]. When higher frequency components are harmonically related to the fundamental (i.e., they fall at integer multiples of the fundamental frequency), they may be called *harmonics*. When higher frequency partials are not harmonically related, they may be referred to as *overtones*.

In referring to frequency, one commonly refers to either radians per second, cycles per second (Hz), or radians per sample. In describing analog systems and real-world sounds, this work primarily refers to frequency measured in Hz, often denoted by f or some variation thereof. When referring to digital systems and models, unless otherwise stated, reference is made to digital frequency, denoted by variations of ω, measured in radians per sample.

Many models described in this work make use of digital delay and other elements. Commonly illustrated via block diagrams, many models are described generally, without regard for how they are realized. For example, many models may have elements labeled "delay," which may be realized by typical integer delay, fractional delay filters, or a combination of the two. Where important, details are given. Additionally, the term "implementation," is used often to refer to specific model designs, methodology, or topology, as opposed to, for example, details of how digital transfer functions are realized in software. Such terminology is abundant in discussion of banded waveguide models.

The physical and physically-inspired models herein are frequently configured as single-input single-output digital filters. Simulating many types of interactions may require more complex, distributed input to models. However, some interactions may be simulated as "injected" input at a point by changes of wave variables or simplifying assumptions. Often, the configuration of inputs and outputs for the models here are such for simplicity in simulation and analysis.

1.5 Contributions

Toward re-sonification of events, we present a method of automatically creating soundscapes to represent the activity of chosen geographic areas, using a database of sounds. Relying on acoustic, semantic, and social information about sounds, this work describes how to design the parameters of a graph-based method of soundscape synthesis such that sounds from or relevant to a selected area are played.

Toward re-sonification of objects, this work proposes and derives alternative topologies for banded waveguide models. When designed correctly, some BWG models proposed here will behave identically to comparable digital waveguide or modal models, subject to linear point-interactions. Subjected to nonlinear bowing interactions, the topologies proposed here demonstrate improved attack characteristics as compared to previous BWG model topologies, with respect to digital waveguide simulations. Numerous implementation details of BWG models are also presented and discussed.

Toward re-sonification of events, a constrained method of input estimation for estimating percussive excitations is proposed. This estimation process assumes the input to a known linear system lies on a signal subspace, and that noisy outputs of this system are observed. Where sounds are known to be percussive (i.e., subjected to an excitation limited in duration), the input estimation method described here may be applied to explicitly consider the limited duration of their excitations. Using this constraint in estimating ex-

citations shows improvements for resynthesis, as compared to unconstrained methods. Results of applying incorrect constraints are presented, as well as methods for recursively calculating optimal input estimates with fixed complexity.

1.6 Outline

The remainder of this report is organized as follows. Chapter 2 presents a review of various methods of representing and understanding sound, with a focus on the perception and production of sounds. The re-sonification of environments is discussed in Chapter 3. Models for the re-sonification of objects are discussed in Chapter 4. Chapter 5 presents methods for modeling of events as interactions with objects. Conclusions are given and future points of research are suggested in Chapter 6.

Additional information is presented in a series of appendices. Appendix A discusses the behavior of poles and resonance in analog and digital propagation models, an important detail in designing and comparing models. The use and design of fractional delay filters in narrowband contexts, such as in banded waveguide models, is examined in Appendix B. Results from simulations of BWG models, described in Chapter 4, are presented in Appendix C, placed as such for ease of reading.

Chapter 2

PRODUCTION, PERCEPTION, AND REPRESENTATIONS OF SOUND

Sound refers to acoustic vibrations; specifically, the term refers to those audible to humans. Generally, this includes vibrations within the frequency range of approximately 20 Hz to 20 kHz. Further, sound is usually heard by humans after propagation through the air. (With audition as an end goal of this work, philosophical questions such as that of the tree in the woods are irrelevant.) The origin of sound lies in the motion of matter, subject to the laws of physics. This motion may occur in anything from a bowed violin string to an electromagnetically-driven speaker. This work is concerned with understanding such production of sound and its perception by humans to work toward ecologically valid re-sonification.

In order for sounds to be stored, reproduced, or altered, they must be represented in some form. Ideally, sonic representations will have meaningful and understandable forms. By "meaningful," it is meant that parameters of the representation explicitly define features related to the production or perception of sounds. Sound signals may be described by virtually any arbitrarily chosen description; representations that consider perceptual or physical properties, however, afford perceptually and physically meaningful transformations. Of primary interest in this work are digital representations of sound, motivated by the abilities and ubiquity of digital processing systems. The taxonomy and classification of models and representations given in this chapter are not exhaustive or unique. Rather, the given classifications are chosen based on the objectives of this work, with an emphasis on perceptually-motivated and physically-based sound models.

In this chapter, perceptual aspects of sound are first discussed, without strict consideration of computation. With re-sonification obviously intended for human use, *how sounds sound* is an important metric. Next, the mechanisms behind sound production

are introduced. Computational models for analysis, transformation, and synthesis are then discussed. In this work, model types are classified as spectral, physical, hybrid (i.e., a hybrid of spectral and physical), or other. Spectral models encompass sinusoidal representations of sound, with many extensions, providing a representation related to the perception of sound. Physical models refer to simulations of the physical phenomena that produce sound and approximations thereof. Hybrid spectral-physical models generally consider the physical production of sound separately for the different resonant or modal frequencies of a sound or object. Other models and representations of sound are mostly omitted for brevity.

For further information about various representations of sound, beyond the scope of this work, detailed exposition of models for sound analysis and synthesis may be found in [2, 14, 19, 33–36] and elsewhere. Evaluation and comparison of various synthesis and modeling methods may be found in [37]. Further details of banded waveguide models, a focus of this work, may be found in Chapter 4.

2.1 Human Perception of Sound

While a time-series representation of varying air pressure at one's ears is effectively sufficient to reproduce sounds without loss of information (neglecting any effect of the other senses or body parts that may be affected), such is far removed from how one might describe such signals. In audition, sounds are subject to physiological perception and cognition. That is, sound waves stimulate organs of the ear which act as transducers, producing neurological signals that are in some way observed by the brain. This process is not wholly understood, but centuries of study have provided useful insights, including knowledge that has been applied to re-sonification [21, 38–42]. This section provides a brief overview of certain aspects of sound perception, as currently understood.

2.1.1 Hearing: Anatomy, Physiology, and Psychoacoustics

To understand audition in humans it is perhaps best to first consider the physical processes behind it. Sound arriving at the ear first travels down the auditory canal, a small tube connecting the ear's opening to the eardrum, also called the tympanic membrane. The eardrum is mechanically coupled to a series of three bones, the *malleus*, *incus*, and *stapes*, also referred to as the hammer, anvil, and stirrup, respectively. These bones transfer vibration to the cochlea, a spiraled tube containing two fluid-filled cavities separated by a structure called the basilar membrane. Vibrations from sound travel along the basilar membrane, exciting hair cells that are sensitive to varying frequencies. The motion of these hair cells then produces nerve pulses that are sent to the brain. (This description and further details may be found in [43].)

As the hair cells within the cochlea are responsive to varying frequency bands, the ear is often loosely interpreted as performing time-frequency analysis of signals [19]. This is a significant motivation behind spectral modeling and the techniques used in many audio codecs [38]. The ear does not function as a "perfect" spectral analyzer, however. Due to the physical response of the ear's organs and the brain's interpretation of auditory nerve signals, perception is dependent on a sound's frequency, amplitude, and transient behavior. For example, to an "average" person, sinusoidal signals at 100 Hz and 1000 kHz at sound pressure levels of 30 dB and 40 dB, respectively, will be perceived to be of approximately equal *loudness* [44]. The perceived loudness of signals differs among people, and is highly dependent on frequencies present in a signal; Fletcher-Munson equal loudness contours present a popular example of quantified loudness perception [44]. Further, slight changes in the frequency or amplitude of a signal may be imperceptible (if they fall beneath the threshold of the *just noticeable difference*) [32].

Masking is another significant phenomenon in audition. Masking is the process in which one sound is not audible due to the presence of another [38]. Generally, this refers to signals within a certain temporal or spectral range of each other. Louder signals generally mask softer signals, and lower frequency signals may mask higher frequency signals when within a *critical bandwidth* of one another [38, 43]. These signals need not be simultaneous, with masking signals having an effect briefly before their onset and for some time after sounding [38]. Masking phenomena has been successfully exploited in lossy audio codecs, where masked components of a signal are removed, providing significant data reduction without necessarily causing significant loss of intelligibility or perceived quality [38, 41].

The above phenomena and others in audition are commonly used to emphasize the "important" parts of sounds in analysis and synthesis. With a focus on allowing robust transformation, however, such phenomena is not explicitly considered in analysis procedures herein, as perceptually insignificant components may be transformed into significance. Additionally, sufficiently accurate reproduction or simulation of sounds may achieve perceptual fidelity. Thus, the intention of the preceding information is to emphasize the general significance of sounds providing physiological stimulation in time and frequency. Note, though, that specific applications may warrant perceptually-based design [38].

2.1.2 Listening: Acoustic Ecology and Cognition

The above phenomena have been described primarily in terms of distinct sinusoidal or narrow-band components. While this provides insight and has proven useful in coding and modeling, most people do not describe sounds by listing frequencies, amplitudes, and bandwidths. Rather, sounds are typically described by the objects from which they originate, the interactions that create them, the places where they can be heard, or

other features. This describes *grouping* or *stream segregation*, where people perceptually group "the parts of the neural spectrogram that go together," [45].

To examine the ecological perception of sound, we consider the concept of *soundscapes*. Pauline Oliveros defines a soundscape as "all of the waveforms faithfully transmitted to our audio cortex by the ear and its mechanisms," [46]. To further specify the types of sounds within soundscapes, Schafer classifies sounds as either *keynote sounds*, *signals*, or *soundmarks* [47]. Keynote sounds are those sounds generally perceived as background sounds, a role informed, like that of the other classifications, by a sound's relevance to an individual. (Indeed, the perception of and reaction to many things are dependent on relevance to individuals – consider, for example, the role of a discarded Coke bottle in *The Gods Must be Crazy* [48] or the utterances of *Prisencolinensinainciusol* [49].) Signals and soundmarks are those foreground sounds that are consciously observed, with soundmarks identified as sounds unique or especially important to a community, analogous to local landmarks. With this taxonomy in mind, we then consider how foreground sounds are perceived.

Bregman states that sounds are "created when things of various types happen" [45]. Few, if any, seem to disagree with this point, but the perceptual classification of individual sounds is contentious, both in terms of what constitutes an individual sound and how it is perceived. Some postulate that in human audition, similar to vision, components of sounds heard are organized (based on shared characteristics or prior knowledge), and individual auditory objects (that represent the source of the sound) are perceived [42, 50]. That is, when one hears a sound, the object from which it originated (or a similar or familiar object) may be perceived in some sense, along with its various physical properties, such as size and material. Alternatively, some consider sounds perceptually observed as the events (interactions between physical matter) that create sonic

vibrations [51]. Yet others, however, refer to the perception of both objects and events in sonic environments [45]. Indeed, this is even recognized in the object-centric work of [50].

When faced with sounds that are purely synthetic, unfamiliar in some way, or even generated from familiar object-event interactions, one may further perceive and describe various sonic qualities by semantic descriptions or even imitation, vocal or otherwise. Perceived qualities include spectral, temporal, physical, and timbral percepts, among others [34]. Thus, sounds are often described, for example, as "bright", "short", "metallic", or "brassy." Though these descriptions may not have a one-to-one correspondence to any specific object or event model, such qualities arise from physical and spectral properties of objects or from interactions.

In this work, the dual perspective of perceived objects and events is considered, as full knowledge of physical objects and the events (or interactions) between them is sufficient to describe and, with accurate models, reproduce their sounds.

2.2 Physical Production of Sound

This section describes the creation of sounds through the vibrations of objects, which may be described by elementary Newtonian mechanics. Given the size and speed of audible objects, both built and encountered, consideration of quantum or relativistic effects is not necessary [2] in this work.

As with other fields of study, the systems described here are simplified models, ignoring many nonlinearities, certain sources of loss, and other difficult-to-quantify phenomenon. Thus, while the physical cause of sound production is discussed in this section, one may alternatively refer to the systems described here as analog physical models, in contrast to digital physical models.

2.2.1 Preliminaries and Assumptions

Much of the focus in analyzing and modeling vibrating systems is on *distributed* systems, as opposed to *lumped* systems. A distributed system is one in which the wavelengths of disturbances are small, such that the system's behavior is space-dependent. Lumped systems consider disturbances of longer wavelengths or at a point. A linear electric circuit operated at DC is a classical example of an electrical lumped system, whereas a coaxial waveguide may be considered a distributed system for RF signals.

Here, many objects are referred to as one-dimensional or multi-dimensional. On atomic scales and above (smaller scales are neglected here) matter occupies three-dimensional space, but many systems are effectively of smaller dimension at the scales considered. In the case of the ideal string of Section 2.2.3, for example, it is assumed that the string has a very small diameter; specifically, its diameter is considered small in comparison to all wavelengths of interest (i.e., those within the audible or simulated range). Similarly, a metal plate of a relatively small, finite thickness may be referred to as two-dimensional. Such assumptions are applied at times for reasons of simplicity. Thus, references to the dimension of objects should not necessarily be interpreted literally (i.e., in a *literal* literal sense).

In the following discussions of sound production, the focus is on describing the vibration of objects. The propagation through air and reverberation of sounds are primarily ignored. Such phenomena play an obvious role in how an object's vibrations are acoustically observed, but for simplicity, they will not be considered in much of this work. Also, the focus is on solid objects, as opposed to acoustic air tubes, cavities, etc., in which sound may be produced in a similar manner by propagating waves, or via nonlinear fluid dynamics [52].

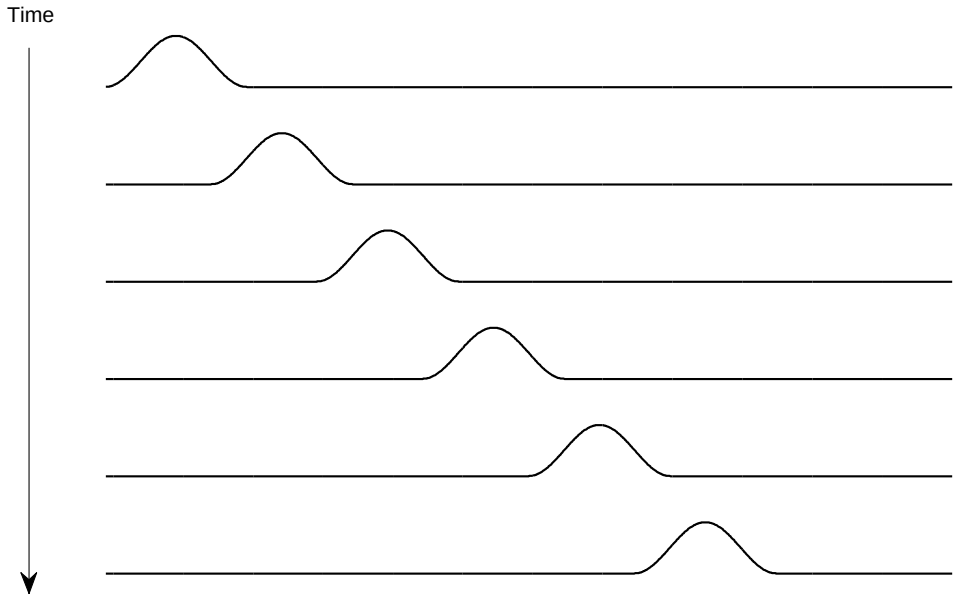


Figure 2.1: An example traveling wave, without loss. Note that the shape is maintained, but travels to the right as time progresses.

2.2.2 Waves

Waves play a critical role in describing the vibrations of objects. Of particular interest are *traveling waves*. These are disturbances, or changes, that propagate in some medium. It is important to note that this does not refer to movement of matter in the direction of propagation – a tsunami is not entirely composed of water molecules from the wave's origin. Rather, a wave is a disturbance moving in space and time; this disturbance then causes motion, compression, torsion, a change in electric field, or some other change [53]. Figure 2.1 shows an example wave, traveling without change in the right direction.

In quantifying waves, one may consider different *wave variables*. For example, in a water wave, one may measure the height displacement of water along the wave, or alternatively, the velocity at which the water rises and falls. Such are not the only wave variables that may be considered in water or other media. Regardless of choice of wave

variables in analyzing or simulating vibrating systems, the behavior of waves is generally similar. One may convert between different wave variables, such as displacement and velocity, through differentiation and integration; in such cases, simulations of or choice of variable for the system of interest should be altered to consider propagation of appropriate wave types [2].

Beyond choice of wave variables, wave phenomena vary in the types of local interactions they elicit. Transverse waves cause displacement perpendicular to the direction of the waves' travel. Waves may also be longitudinal, causing displacements in the direction of travel, resulting in compression and rarefaction of the medium. Further, twisting motion can be caused by torsional waves. Though these wave types may all occur in the same medium, they need not have the same properties, such as speed of travel. Thorough discussion of differing wave types may be found in [54].

Waves are generally subject to various alterations as they travel, decreasing in magnitude and changing shape and direction. Such alterations account for the lack of purely periodic sounds in the vibration of real-world objects. The causes of these processes are discussed in the following sections. For more about general wave phenomena, see [55].

2.2.3 The Ideal String

The lossless, ideal, and elastic vibrating string is perhaps the simplest example of a distributed system from which to understand essential properties of physical models. For such a string, under tension, K, with mass density, ϵ, its transverse displacement, y, at time, t, and position along the string, x, satisfies the one-dimensional wave equation,

$$K\frac{\partial^2 y}{\partial x^2} = \epsilon\frac{\partial^2 y}{\partial t^2}, \tag{2.1}$$

assuming the displacement is relatively small. (Here, we are considering transverse displacement in a plane, neglecting the other dimension of transverse displacement, as well as torsional rotation and longitudinal compression of the string.) Eq. (2.1) describes how,

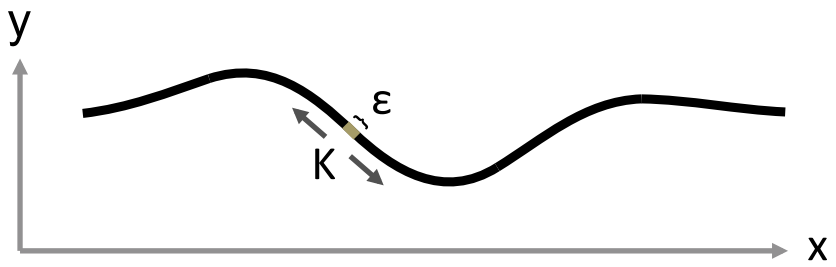


Figure 2.2: An example section of a displaced string under tension K, with mass density ϵ.

as a result of the string being under tension, the acceleration of the string at each point is proportional to its curvature at that point. An example section of string is shown in Figure 2.2.

The solution of (2.1) was shown by d'Alembert to have the form,

$$y(x,t) = y_r(x - ct) + y_l(x + ct), \tag{2.2}$$

where $c = \sqrt{K/\epsilon}$ [56]. Examination of (2.2) reveals this general solution to be interpretable as the sum of a right-going wave, y_r, and a left-going wave, y_l. That is, for some time, $t_1 > t_0$, $y_r(x - ct_1)$ will appear the same as $y_r(x - ct_0)$, but shifted to the right along x; the same behavior follows for y_l, but to the left. This wave-based interpretation then reveals c as the propagation velocity of the transverse displacement waves.

The behavior of an ideal string is further defined by its boundary conditions: the conditions that describe the ends of the string. (We can mathematically realize an infinite string under tension, but such is obviously not the case in reality.) If a string of length L is rigidly held in place at its two endpoints ($x = 0$ and $x = L$), then we have the boundary

conditions,

$$y(0,t) = 0 \tag{2.3}$$

$$y(L,t) = 0. \tag{2.4}$$

This implies that

$$y_r(0,t) = -y_l(0,t) \tag{2.5}$$

$$y_r(L,t) = -y_l(L,t), \tag{2.6}$$

which may be interpreted as the right-going and left-going waves at the ends of the string inverting and then reflecting, traveling in the opposite direction. Other boundary conditions exist, such as free ends that cause non-inverting reflections [52], but they will not be considered in this example.

As a result of its boundary conditions, the rigidly-terminated ideal string exhibits periodic behavior. If the motion of the string at any point (where the motion is due to some initial displacement) is observed, it will be a periodic function with fundamental frequency,

$$f_0 = \frac{c}{2L}, \tag{2.7}$$

where c is measured in meters per second and L in meters. This periodic oscillation results from disturbances traversing the string and arriving at their original position every $1/f_0$ seconds. This is a simple case of a closed wave train, further discussed in Section 2.2.8.

An alternative expression for the solution of the wave equation, where there are rigid terminations, is

$$y(x,t) = \sum_{k=1}^{\infty} A_k \sin(\beta_k x)\cos(\omega_k t + \phi_k), \tag{2.8}$$

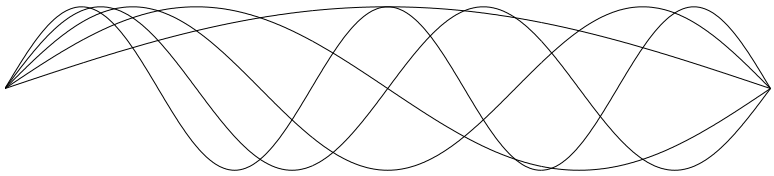


Figure 2.3: The first five mode shapes of a rigidly-terminated ideal string.

where

$$\beta_k = \frac{k\pi}{L}, \tag{2.9}$$

$$\omega_k = \frac{ck\pi}{L}, \tag{2.10}$$

and A_k and ϕ_k are constants, dependent on initial conditions of the string [57, 58]. This solution provides a *modal* view of a lossless, rigidly terminated ideal string, discussed further in the next section.

Another simple acoustical model for sound production is the acoustic tube. While tubes will not be discussed in depth, it should be noted their behavior is in many ways similar to that of a string, with air pressure and other fluid variables exhibiting wave behavior. See [2] for further discussion about strings and the derivation of traveling waves in tubes.

2.2.4 Modal Decomposition

Modal analysis decomposes the vibrations of objects into several components, called *modes*, that vibrate at some frequency and, in lossy systems, decay at some rate [59]. Each mode has associated modal data: frequency, decay, and shape. The frequency and decay of each mode characterize its damped oscillatory behavior. Mode shapes describe the extent to which each portion of an object vibrates at each mode. From the sum of all modes, one may reconstruct structural vibrations. Similarly, the excitation of modes due to an initial displacement may be found by the extent of the mode shapes

at the displaced portions of the object [59]. Modal data may be derived from mathematical descriptions of structures, but it is often estimated empirically, proving useful in analyzing the vibrations of bridges and other structures [34].

In the example of the lossless, rigidly-terminated ideal string, (2.8), the term $\sin(\beta_k x)$ describes the mode shape associated with the modal frequency ω_k. The first five such mode shapes are shown in Figure 2.3. Note that this example has no decay – or an infinitely long decay – as there are no losses. The constants A_k and ϕ_k describe the amplitude and phase, respectively, of the vibrations of each mode, as determined by initial conditions.

Modal analysis provides a spectral-based view of vibrations. As such, it is the basis for many hybrid spectral-physical models [6, 7, 31], providing meaningful spectral decompositions of vibrating physical objects. These hybrid methods are discussed in Section 2.5.

2.2.5 Losses

The ideal string described above will, subject to some initial displacement, oscillate indefinitely. This is due to the lack of any loss in the system of (2.1). Strings in the real world, however, do not vibrate forever; various sources of energy loss contribute to the decay of vibrations. Where losses along the string are uniform and proportional to velocity, they may be modeled [2, 60] by a modification of (2.1):

$$K\frac{\partial^2 y}{\partial x^2} = \epsilon\frac{\partial^2 y}{\partial t^2} + \mu\frac{\partial y}{\partial t}. \tag{2.11}$$

Losses of this type result in the exponential (temporal) decay of traveling waves; this may equivalently be seen as propagating waves undergoing a constant multiplicative loss for every traversal of some chosen length of string [2]. The first-order term of (2.11) may be viewed as a simplification of losses attributable to heat dissipation [60]. These losses are

frequency-independent and only have an extremely minute effect on modal frequencies (cf. Appendix A).

Other sources of loss in strings, such as the acoustic radiation of energy, are generally frequency-dependent, resulting in the attenuation of traveling waves being viewable as a frequency-dependent filtering process [2]. This results in the overtones of a string's vibrations decaying at differing rates; generally, higher frequencies are subject to faster decay. (See Figure 2.4 for an example of wave traveling in a medium with frequency-dependent loss.) Such phenomena is often modeled by introducing additional terms to the string's equation of motion. These additional terms are typically odd-order partial time derivatives of transverse displacement with respect to time [60–62] or possibly mixed-derivatives [1, 2]. From a physical standpoint, justification of such terms may be "tenuous," but they are responsible for "perceptually important variations in damping rates," [1]. (See Appendix A for further discussion and details of losses in such systems.)

Losses such as those described above may complicate analysis and modeling of musical instruments but contribute to their appeal and utility. In [30], Steiglitz notes, in passing, the departure from purely periodic behavior as a source of interestingness for musical sounds. Yielding terminations, such as the bridge of a guitar, can further enrich the instrument's sound by behaving differently for horizontal and vertical vibrations, leading to a "chorus effect" [2]. Such losses also play a critical role in acoustic instruments, as energy lost to the air, terminations, and body of an instrument is the source of the radiated sound.

2.2.6 Dispersion

In the case of the ideal string, it is considered to be flexible and lack any stiffness; i.e., when not under tension or subject to other forces, it will not change shape. Many strings, however, exhibit stiffness. Consider the thickest wound strings of an electric

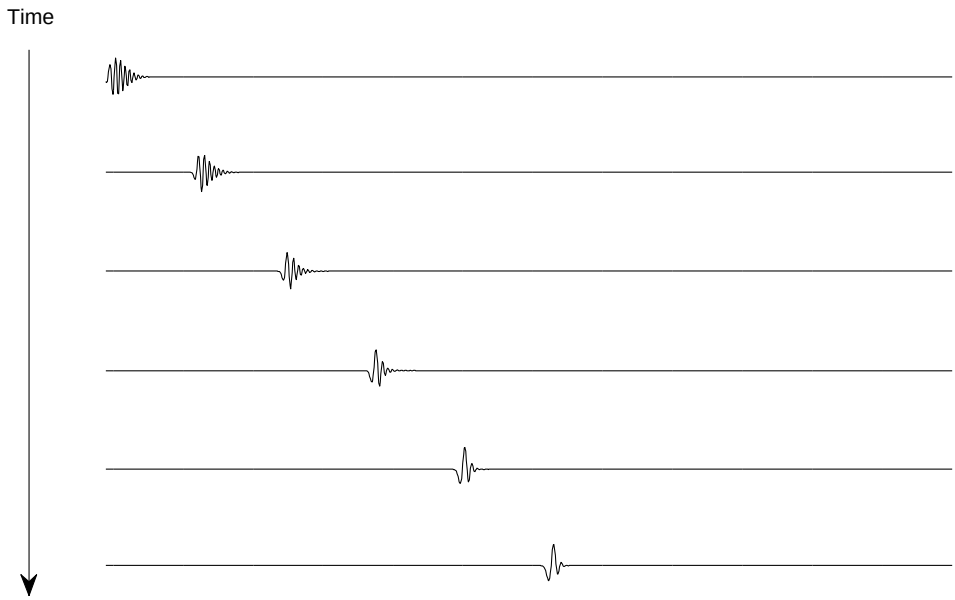


Figure 2.4: A traveling wave undergoing frequency-dependent losses. Note that the high-frequency components with shorter wave lengths are attenuated as time progresses and the wave shape is smoothed. (The wave magnitudes are scaled for comparison.)

bass guitar, for example. Stiffness creates a restoring force in response to deformation. In some objects, such as stiff bars, stiffness provides the only restoring force responsible for traveling waves [57]. Stiffness in strings and other objects results in *dispersion*, the changing of traveling wave shapes as the various frequency components separate.

Unlike tension-borne restoring forces, restoration due to stiffness causes frequency-dependent behavior. Specifically, higher frequencies travel with a faster propagation speed. One may intuit this phenomenon by considering that high frequency waveforms have greater curvature; this increased bending at higher frequencies then results in an increased restoring force. Thus, the propagation speed in a stiff system increases with frequency. As such, propagating waves in a stiff system disperse, with higher frequency components traveling faster and "outrunning" lower frequency components. This phenomenon is shown in Figure 2.5.

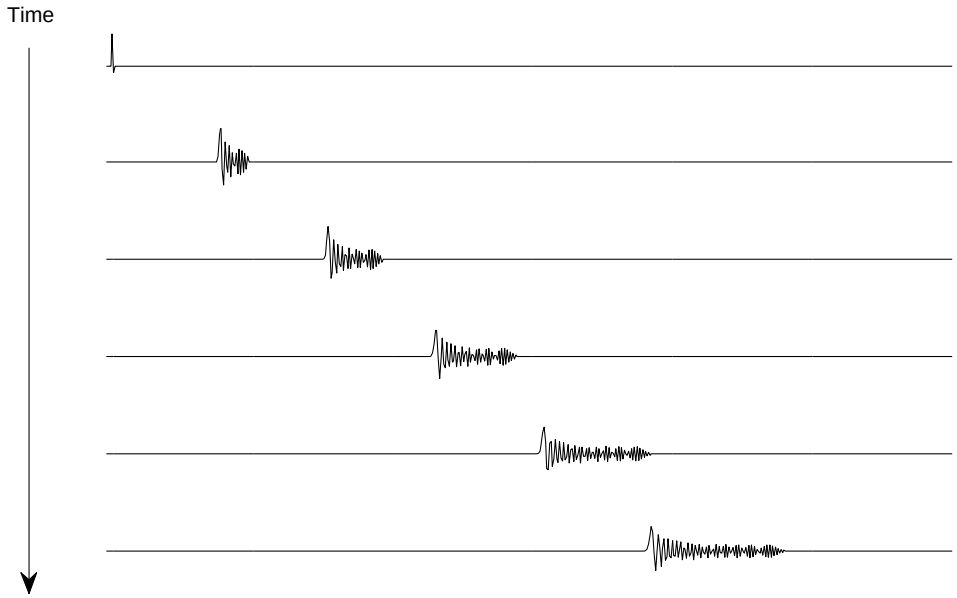


Figure 2.5: A traveling wave displaying dispersion, with high frequency components traveling faster than low frequency components. (Wave magnitudes are scaled for comparison.)

As shown in [57], stiffness introduces higher-order terms in equations of motion. For example, the equation of motion of a bar is

$$\frac{\partial^4 y}{\partial x^4} = -\alpha \frac{\partial^2 y}{\partial t^2}, \tag{2.12}$$

where α is a constant, dependent on the physical properties of the bar [57]. Similarly, stiff strings may be modeled by adding a fourth-order term to (2.1) to account for the restoring force due to stiffness [2]:

$$K \frac{\partial^2 y}{\partial x^2} - \eta \frac{\partial^4 y}{\partial x^4} = \epsilon \frac{\partial^2 y}{\partial t^2}. \tag{2.13}$$

Here, η is the product of the string's Young's modulus (stress over strain) and area moment of inertia [2]. More accurate models may incorporate further terms [2].

From a modal perspective, stiffness causes inharmonicity. With propagation speed increasing with frequency, the resonant frequencies in a system are "stretched."

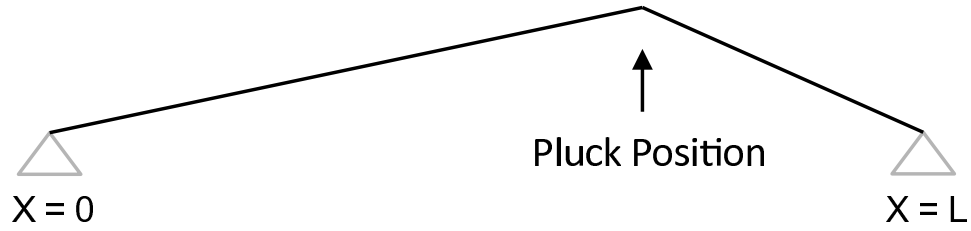


Figure 2.6: The shape of a plucked ideal string with no stiffness before release, assuming an infinitesimally small plectrum.

The spectra of a stiff string, for example, is then inharmonic, with the various overtones increased in frequency. In slightly stiff media, such as guitar strings, the effect may be slight; however, very stiff objects, such as bars, exhibit strong inharmonicity.

Note that stiffness in bars and other systems may also give rise to non-propagating "near-field" oscillations, dependent on boundary conditions and excitation [54]. In propagation models, such as BWGs, these oscillations can only be modeled as propagating waves [7]. With a focus on propagation models, correct modeling of near-field terms is beyond the scope of this work.

2.2.7 Interactions

Systems can only produce vibrations as a result of the presence of energy. This can include potential energy such as that present in the initial displacement of a string under tension, or the kinetic energy of a vibrating string. Various interactions, such as plucking, bowing, or hammering, may *excite* sounding objects, imparting energy into them. There is generally an audible difference in the sound produced by different forms of excitation, and interactions with objects must therefore be understood to properly describe and model sound generation.

Understanding interactions requires considering the physical processes that impart energy into a sounding object. For the plucking of a string, this can be as simple

as considering the string as having an initial displacement, approximately in the shape of two lines from the boundaries to the plectra [2]. This is illustrated in Figure 2.6. Other interactions, such as hammering a string may be seen as introducing a pulse of velocity into the string (at the area of hammering) [2]. For such simple interactions, especially given their short duration, it generally suffices to consider them as directly altering the object's state. More complex, sustained interactions, however, may couple interacting objects more strongly.

Bowing and other friction-based interactions involve highly nonlinear processes with feedback between objects and exciters, complicating their analysis and modeling. Such interactions, however, have driven much of the work in understanding and modeling musical instruments [3, 63–68]. In the particular case of bowing a string, the friction between the string and bow varies in time, dependent on the state of the string and bowing technique; bowed strings are typically described as being in a state of "sticking" to or "slipping" against the bow. Analysis of this behavior [63, 64] has been applied in numerous models of bowed strings [2, 69].

Analyzing interaction with sounding objects, much may be understood from a frequency perspective. Linear systems can only output signals in frequency ranges for which energy is input in the system (or initially present). With many instruments largely characterized by their resonant frequencies and considered effectively linear, they may only produce sounds at frequencies present in their excitations. Thus, it is not surprising that many methods of interaction are wide-band. (Consider the impulsive nature of hammering, for example.) Spectral content, though, may be further influenced by other aspects of interactions, such as position (cf. Section 2.2.4).

2.2.8 *Closed Wave Trains*

A closed wave train is a path of propagation in which a wave of some frequency returns to its point of origin with the same phase and direction of propagation [7, 54, 70]. This "wave train closure principle" [54] describes the physical source of modes (and relatedly, resonance) in objects, providing a link between spectral theory and physical models in terms of propagating waves [7, 70]. This is the motivation for sound synthesis by banded waveguide models.

The simplest example of a closed wave train is the rigidly terminated ideal string (without dispersion or loss). In the ideal string of Section 2.2.3, after some excitation or initial displacement, waves propagate back and forth along the string, arriving at an identical configuration every $2L/c$ seconds. Thus, a mode occurs at the fundamental frequency of the string (2.7), the inverse of the time for waves to make a round-trip traversal of the string. Similarly, modes emerge at the harmonic frequencies of the string's fundamental (2.7), with an integer number of periods occurring every $2L/c$ seconds.

Few, if any, real-world objects exhibit truly lossless or dispersion-free propagation; as a result, their sounds are quasi-harmonic or pseudo-periodic. Still, objects often exhibit strong resonant peaks in the spectra of their sounds, many the result of propagating waveforms that satisfy the "wave train closure principle." This principle holds true in multi-dimensional objects. Somewhat analogous to ray-tracing in graphics [71], wave train closure arises in objects of arbitrary dimension and shape wherever a wavefront may arrive at its origin, with its original phase, after propagating throughout an object, subject to losses, reflections, and/or dispersion. (Graphic ray-tracing, however, does not regard phase.) This motivates propagation modeling for efficiently synthesizing the sounds of various multi-dimensional and stiff instruments [72]. An example path corresponding

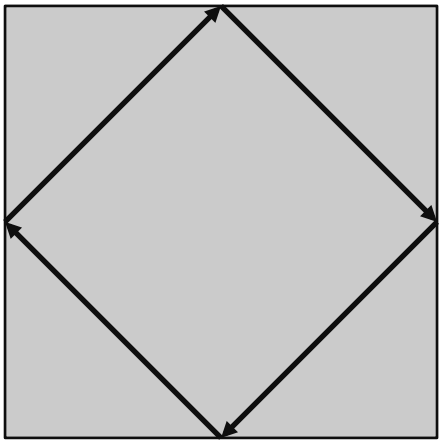


Figure 2.7: An example path on a square plate corresponding to closed wave trains for those frequencies for which waves meet themselves in phase.

to a closed wave train (for the appropriate frequencies that meet themselves in phase) for a square plate is shown in Figure 2.7.

2.3 Spectral Models

Spectral modeling broadly refers to representing signals by a combination of sinusoids with possibly time-varying amplitudes, phases, and frequencies. As the human cochlea responds over time to energy within different frequency bands, spectral models may be considered as coarsely modeling sounds by their physiological perception. As discussed in Section 2.1, the process of sonic perception is more complicated and not wholly understood, but spectral decomposition provides a means of quantifying perceptually important components of sounds. This viewpoint has its roots in the work of Helmholtz [73] and is perhaps best justified by the wealth of synthesis methods employing it [19].

Beyond the physiological motivation for spectral models of sound, practical reasons abound for their use. Fast Fourier Transform (FFT) algorithms [74] allow relatively quick synthesis and analysis in the frequency domain [75]. (Temporally-based synthesis

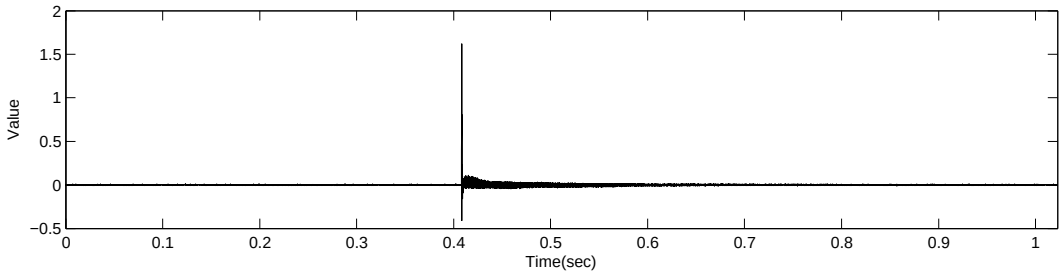


Figure 2.8: The time-domain waveform of a sound with transient, sinusoidal, and noisy components.

is also feasible [76].) With a time-dependent spectral representation, time and frequency shifting and scaling is relatively straightforward [77]. For these reasons and the fact that spectral analysis is widely studied and applied in other fields, spectral modeling of sound has proven both meaningful and useful.

Most spectral modeling methods consider sound in successive frames, i.e., short portions of a signal. These frames are often windowed, being multiplied by a function in time, to smooth discontinuities in the time domain or provide some compromise between spectral and temporal resolution. Such frame-by-frame analysis and synthesis is also found in other sound modeling methods such as granular synthesis [34,78], but here we consider it as a tool for time-frequency analysis, often in the form of a short-time Fourier transform (STFT).

A brief review of spectral modeling methods follows, focusing on systems for analysis, transformation, and synthesis. To illustrate the methods detailed here, we consider an example sound, the time-domain waveform and spectrogram of which are shown in Figures 2.8 and 2.9, respectively. Note that this example sound shows a short, transient burst, followed by frequency-shifting sinusoidal components, all in the presence of noise. For a more detailed history and survey of methods, see [19] and [79].

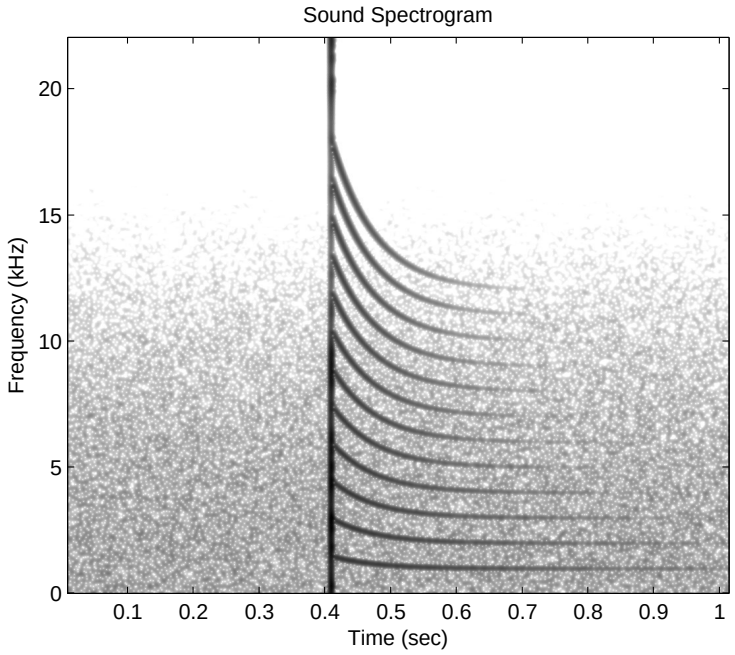


Figure 2.9: The spectrogram of a sound with transient, sinusoidal, and noisy components. Black represents 0 dB and white represents -60 dB and below.

2.3.1 Spectral Modeling Methods

Following the early work of Helmholtz, many spectral-based models have been introduced for analyzing, transforming, and synthesizing sounds [19]. While many early methods use analog electronics, operating in the continuous time domain, much has since been achieved through digital means. This section reviews some of the early digital methods for spectral modeling, typically relying on time-frequency analysis.

Phase vocoders, one of the earliest widely adopted spectral modeling methods, take many forms [80], generally referring to FFT or filter-based analyzers coupled with similarly based synthesizers. Phase vocoders are well-suited for identity resynthesis and time and frequency-based transformations [80–82], but they often do not provide a meaningful representation of sound beyond a simple STFT. Illustrated by many applications [81, 82], phase vocoders can be successfully applied to a wide range of sounds,

but they were originally intended and are perhaps best suited for quasi-harmonic sounds lacking vibrato [83].

Sinusoidal models provide a more meaningful parametric representation of sound signals, as compared to phase vocoders, by detecting sinusoidal signal components and estimating their time-varying properties (i.e., amplitude and phase/frequency) [75, 84, 85]. Such methods and their extensions are generally referred to as Spectral Modeling Synthesis (SMS), among other names. As with phase vocoders, sinusoidal models easily afford time and frequency modifications of sounds [75, 79, 83]. Besides simple shifting and scaling in time and frequency, though, independent manipulation of individual or groups of sinusoidal components is achievable in SMS [83]. This relatively fine level of control enables both arbitrary and informed (e.g., by physical parameters) transformation of sounds. By informed transformation, we mean cases where equally scaling or moving sinusoidal components in frequency or time may not achieve a desired effect; consider, for example, pitch shifting a sung vowel, where the amplitudes of the sinusoidal components should be altered so as to maintain formant shapes. Like any model for representing data, SMS may benefit from any prior knowledge. Examples of sinusoidal components, extracted from the example signal displayed in Figures 2.8 and 2.9, are shown in Figures 2.10 and 2.11.

SMS analysis methods often rely on "peak-picking" in the frequency domain [75], finding maxima in the magnitude spectra of successive frames. Peaks near one another in frequency in successive frames are usually considered as continuations of one another, resulting in "births" and "deaths" of sinusoidal "tracks" [79, 83]. Determining the evolution of the frequency and amplitude of sinusoidal components of a signal is a very active area of research in audio processing. Changes in frequency and amplitude may be found by estimating modulation parameters [86, 87] or by using probabilistic models [88].

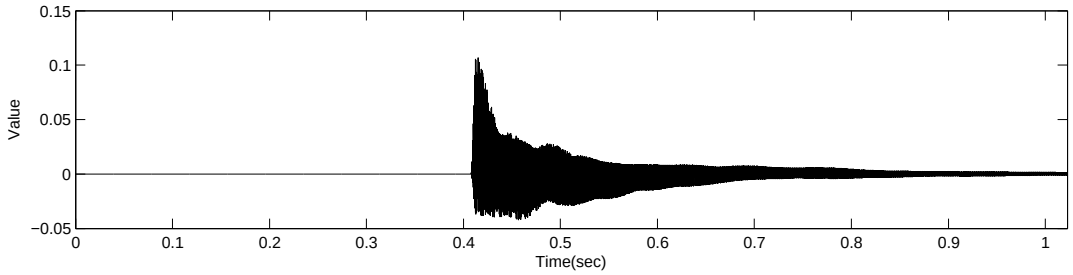


Figure 2.10: The time-domain waveform of the sinusoidal components.

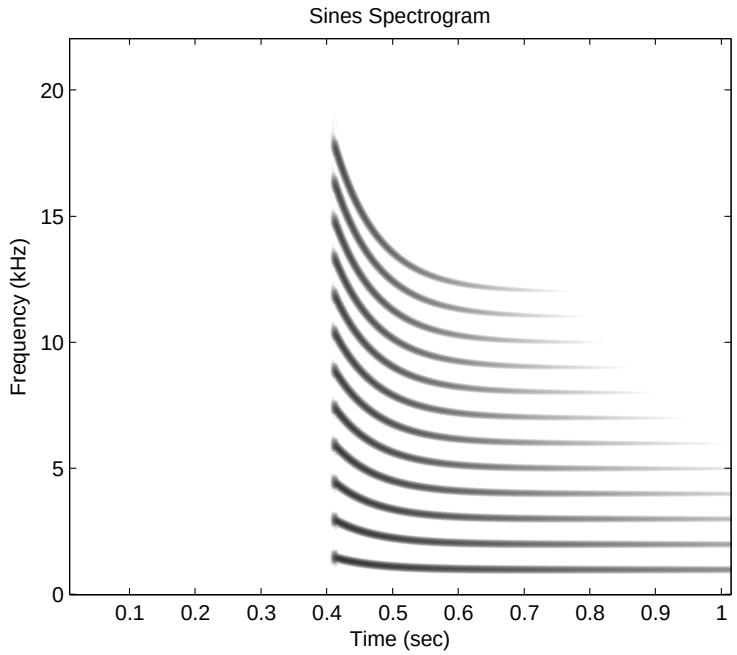


Figure 2.11: The spectrogram of the sinusoidal components. Black represents 0 dB and white represents -60 dB and below.

Generally, spectral peaks are easily identified for narrow-band, sustained sinusoidal components, as they are usually prominent in magnitude spectra [75]. Any wide-band components of sound, however, such as "noise" or percussive transient sounds, are difficult to identify and synthesize by simple sinusoidal representations [79, 83]. This motivates extensions of SMS, detailed in the following sections.

2.3.2 *Noise Modeling*

In [79] and [83], it was proposed that sinusoidal modeling be supplemented by noise modeling. Noise modeling separates a sound into deterministic (sinusoidal) and stochastic (noise) components. Following a sinusoidal modeling stage, noise modeling attempts to compactly describe the residual of a signal when the sinusoids are removed [79, 83]. As the noise is considered stochastic in nature, it may be represented by its magnitude spectra, and perceptually re-synthesized by filtering white noise or generating random discrete Fourier transform (DFT) realizations [40, 79, 83, 89]. An example of a modeled noisy component of a sound, extracted from the example signal displayed in Figures 2.8 and 2.9, is shown in Figures 2.12 and 2.13.

Separate representations for the sinusoidal and noisy portions of a sound enable similar or independent transformations for resynthesis. Meaningful transformation may rely on understanding the nature of the signal components [83]. In [90], it is suggested that noise components of sound not be altered in pitch-shifting operations, due to their atonal nature; however, with no prior knowledge of the noisy portion's mechanism of generation, such transformations may yield undesirable sounds. Additionally, with just a sines and noise model, any quickly decaying components (sinusoidal or otherwise) that are not stochastic in nature may be inappropriately modeled as noise, due to their wide-band spectra.

2.3.3 *Transient Modeling*

Noted in early work for sines + noise SMS [79], many sounds contain a short-lived energetic attack, referred to as the *transient* portion of a signal. The short duration of transient signals causes a large wide-band distribution of energy in the frequency domain; this disqualifies sinusoidal modeling for efficient or meaningful representation of transients [91]. Modeling of transient signals as noise can lead to "smearing" effects when

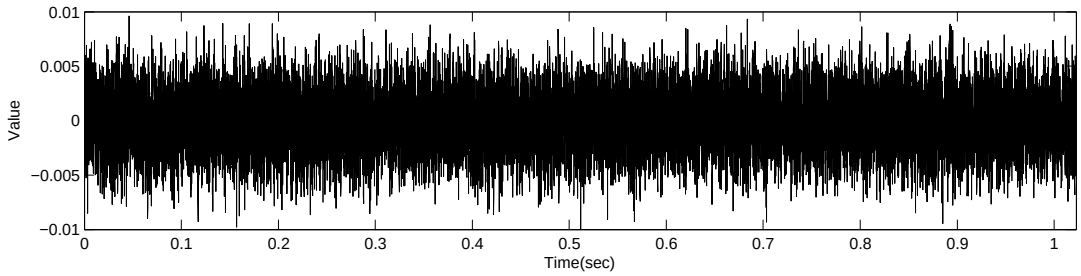


Figure 2.12: The time-domain waveform of the modeled noise component.

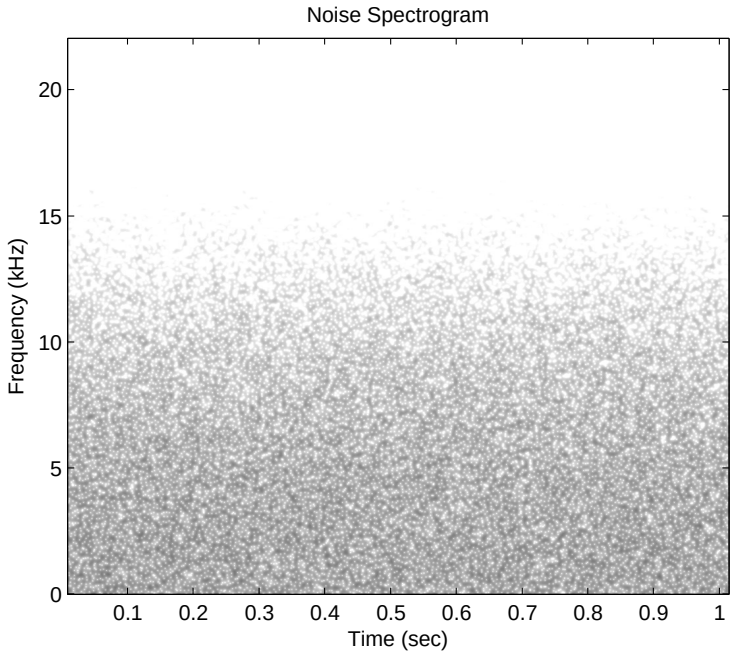


Figure 2.13: The spectrogram of the modeled noisy component. Note that in this example, the spectral noise properties do not change over time. Black represents 0 dB and white represents -60 dB and below.

time-stretching a signal [92]. Transients may be estimated from residual signals after normal SMS analysis, but noise will be retained in such representations [92]. Therefore, a separate and meaningful representation of the transient portions of a signal is needed to allow robust re-sonification of general sound signals within the SMS framework.

Transient Modeling Synthesis (TMS), introduced in [92], attempts to provide a parametric representation of transient signals, separate from noisy and sinusoidal components. TMS effectively applies SMS with a frequency-domain signal as input, as opposed to a time-domain signal used with ordinary SMS. That is, a frequency-domain representation of a signal is obtained, such as a discrete cosine transform (DCT), and SMS is applied to this representation [40, 90, 92]. This results in peak-picking occurring in a pseudo-time domain, obtained by taking a Fourier transform of a signal's DCT. By performing a DCT on larger frames (on the order of 30 to 60 sinusoidal modeling frames [90]) after the removal of the sinusoidal components, the transient portions will create slowly-varying sinusoidal signals in the DCT domain. These DCT signals are then modeled by the normal SMS procedure [40, 90, 92]. An example transient component, extracted from the example signal displayed in Figures 2.8 and 2.9, is shown in Figures 2.14, 2.15, and 2.16.

Transient modeling has also been applied in "sines + transient + noise" [39, 89] and "transient $\star$ sines + noise" [27, 91] schemes . In sines + transient + noise, transient regions and sines + noise regions are separately modeled; transient regions are represented by transform coding, quantized by perceptual criteria [39, 89]. The transient $\star$ sines + noise scheme uses a source/filter approach to model transients by the necessary information to drive oscillators to reproduce them, and this method may benefit from sources of prior knowledge [91].

Explicit modeling of transients allows more robust transformation of signals, as compared to sines + noise modeling. When performing time-stretching, the durations of

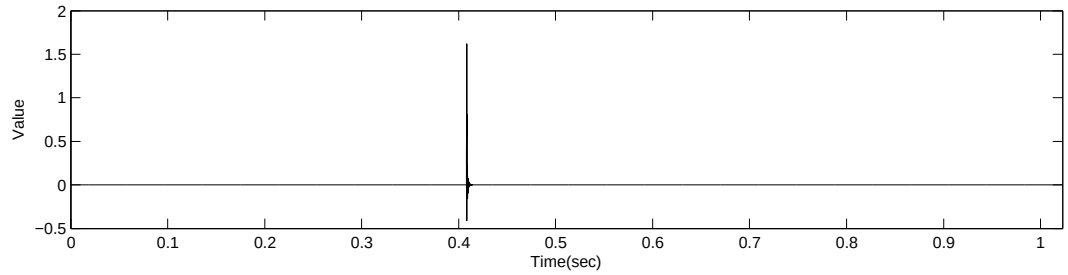


Figure 2.14: The time-domain waveform of the transient component.

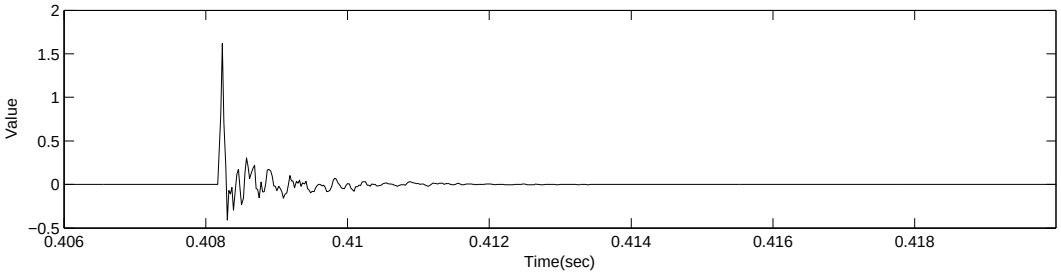


Figure 2.15: Detail of the time-domain waveform of the transient component.

transient regions should generally be kept constant (to prevent "smearing"), with their offsets properly aligned with time-stretched sinusoidal and noisy components [40, 90]. As transients have no distinguishable pitch, they should not be modified when a signal is frequency-shifted [40, 90]. This approach to transforming transients is largely informed by the physical origins of transients, as they are often primarily attributable to the excitation of a physical system [91]. This will be further discussed in Section 2.4 .

2.4 Physical and Physically Inspired Models

Physical modeling (PM) encompasses a wealth of methods that simulate the real-world physical production of sound in some way. In general, sound is produced by the vibrations of a system subject to some external stimulus. In digital modeling approaches, this usually motivates a *source/filter* or *exciter/resonator* approach, separating the model of an object and the interactive events with it. Where the state and configuration of an

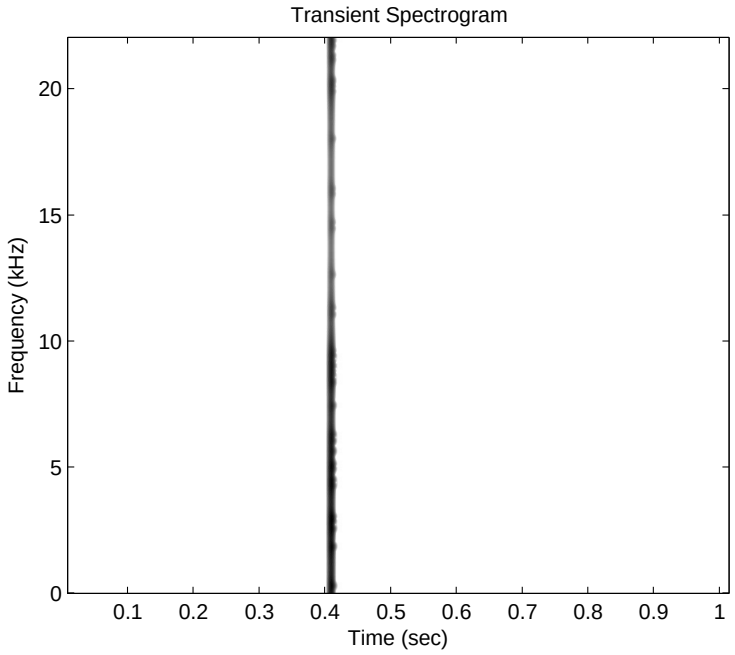


Figure 2.16: The spectrogram of the transient component. Black represents 0 dB and white represents -60 dB and below.

object is known, along with a description of any stimuli, resultant vibrations may be sufficiently described by classical mechanics. In digitally implementing physical models where the dynamics of the corresponding real-world objects are known, two chief obstacles arise. First, the dynamics of any object and interactions with it must be digitized. Second, PM methods should be efficient so as to prevent unnecessary computation and, ideally, allow real-time sound synthesis. PM is typically further complicated, however, by the lack of knowledge of the dynamics of various objects.

Many approaches to PM are concerned with modeling either the vibration of objects or the propagation of waves within them. With the method of sound production in an object explicitly modeled, PM extends to objects ranging from simple (e.g., guitar strings) to complex (e.g., cymbals) [2]. Interactions within and between objects may be modified, allowing simple wave propagation or highly nonlinear behavior. This is a

key strength of PM, as it allows intimate control and manipulation of virtual sounding objects and musical instruments.

Compared to other sound models, PM easily allows alteration of simulated interactions with virtual objects. PM has been applied in modeling many different interactions, including bowed strings [3] and hammered piano strings [93]. A further advantage of PM lies in the ability to alter the perceived properties of virtual objects, such as size, material, and shape [42]. Since PM parameters correspond to physical properties, their manipulation corresponds to altering physical properties of the modeled object. Interestingly, such model parameters may be changed so as to simulate unrealizable or impractical objects, making otherwise infeasible instruments available to musicians and composers, as buddingly evidenced in David A. Jaffe's "Silicon Valley Breakdown" [94].

This section describes various PM methods, many of which rely on approximation, linearization, or other idealizations. Thus, many may be considered as pseudo-physical, physically inspired, or perhaps models of ideal or digitally convenient systems. PM is extensively detailed in [2] and the references therein. Methods of controlling and interacting with these models is also presented; further approaches to PM control and interaction may be found in [3] and elsewhere.

2.4.1 Early and Miscellaneous Physical Models

Physical models for sound synthesis have been used since before the proliferation of digital technology [2]. Various digital models have been realized, for synthesizing voice, musical instruments, object collisions, and other domains. Here, we review some relevant early work in PM, as well as other methods not focused upon in this work. Note that many of these models are still in use.

The Kelly-Lochbaum (KL) vocal tract model is considered a candidate for the first digital physical model for sound synthesis [2]. The KL model simulates traveling waves

in the vocal tract to synthesize singing. Viewable as a digital waveguide model, it models the vocal tract as sections of cylindrical tubes. The excitation of KL models may be a succession of impulses or a simulation of waveforms from the glottis, depending on the fidelity desired. More information on KL models, extensions, and singing synthesis may be found in [95, 96].

Finite difference and finite element methods for physical modeling allow approximate solutions of PDEs to be found by approximating differential equations by difference equations. This has been implemented in [97, 98]. In general, finite element methods can be computationally expensive, preventing real-time use. Note however, that finite difference schemes can be equivalently and more efficiently implemented using digital waveguide methods [2, 99, 100]. Finite difference models may be excited in the same manner as digital waveguides, discussed in Section 2.4.3.

Alternative approaches to physical modeling by numerically solving PDEs include the functional transformation method. The functional transformation method represents a system of PDEs by a transfer function model, using the Laplace transform and similar transforms for space [101]. Using a bilinear transform or other method of converting continuous systems to discrete systems, the transfer function model may be discretized [101]. Application of appropriate inverse spatial transformations and an inverse z-transform then results in a discrete solution of the original system of PDEs [101]. This method has been used to simulate varying objects including circular membranes [102] and tubular bells [103].

McIntyre Schumacher Woodhouse (MSW) synthesis, named after its authors, is one of the early successful methods for modeling bowed strings and wind instruments [65, 66]. MSW synthesis effectively characterizes a linear system by its impulse response. In the case of a violin, this includes propagation of disturbances across the string and re-

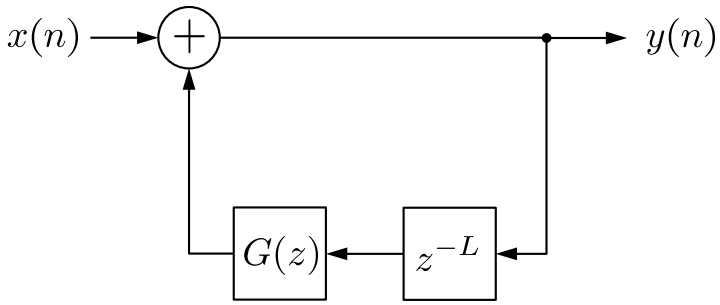


Figure 2.17: A digital filter implementation of the Karplus-Strong algorithm. (This structure is also referred to as a string loop.)

flection at the terminations. The input to the linear system model is determined by some excitation mechanism; in the case of a bowed string, this is a nonlinear model, relying on feedback to model dynamics between the bow and string [63, 64, 66]. MSW synthesis distinguished itself from many other contemporary approaches for sound synthesis in its explicit consideration of time-domain simulation [34].

2.4.2 The Karplus-Strong Algorithm

Proposed for producing string or drum-like sounds in [104], concurrently with extensions [105], the Karplus-Strong (KS) algorithm is a simplified physical model for sound synthesis. (Depending on the implementation and topology, it may be more appropriate to refer to a KS model as pseudo-physical or physically inspired.) Details vary across implementations, but the algorithm primarily consists of filtering a burst of noise with a filtered delay loop. An illustration of a filter implementing the KS algorithm, also known as a *string loop* [4], appears in Figure 2.17, where z^{-L} represents an integer delay of L samples, $G(z)$ represents a (usually low-pass) filter, and $x(n)$ is assumed to be a signal of L randomly selected sample values followed by zeros. The frequency response of this KS filter is similar to that of a comb filter, with nearly harmonic peaks; however, their

width and amplitude are affected by the filter in the feedback loop. (As are the resonant frequencies, usually to a minor degree, as described in Appendix A). The time-domain effect of this is that the waveform is pseudo-periodic, with successive pseudo-periods being filtered versions of each previous one. From a modal perspective, the magnitude of the filter's frequency response at each mode affects the decays of the respective modes. The choice of noise as the initial excitation is motivated by the desire to provide energy at all of the resonant frequencies in the comb-like frequency response of the filter.

Many extensions to the KS algorithm have been proposed, mostly introduced in [105]. Placing a fractional delay filter, such as a first-order allpass filter, in the feedback loop allows finer tuning of the fundamental frequency of the KS filter; otherwise, the fundamental is limited to frequencies corresponding to a period approximately equal to the phase delay of the loop filter plus an integer number of samples. Further, a filter without linear phase response, such as an allpass filter, may be introduced to the feedback path to model dispersion due to string stiffness. (See Appendices A and B for more information on the tuning of such models.) Filters may be introduced before the feedback path, as well, filtering the input; feedforward comb filters are often used as such, to simulate the effect of plucking position. Descriptions of these and other filters to modify the KS algorithm may be found in [2, 105], describing the Extended Karplus-Strong (EKS) algorithm.

KS synthesis is a historical and conceptual precursor to digital waveguide models, discussed below. It may be seen as an efficient implementation of MSW synthesis [105], but it also represents a digital waveguide model, separated into conceptual components. The link between KS and digital waveguide models is further discussed in [4] and the following section.

2.4.3 Digital Waveguides

Digital waveguide (DWG) models are efficient digital filter structures for propagation modeling. Originally developed for reverberation simulation [106], DWG models have been used to simulate violins, guitars, clarinets, and numerous other instruments [2]. The efficiency of DWG models largely arises from assumptions of the linearity of modeled objects, sparsely located interaction and observation, and the subsequent ability to apply aspects of linear system theory. This section introduces DWGs, after [2], illustrating how they model propagation, and how various assumptions and approximations allow efficient sound synthesis.

Consider the ideal string described in Section 2.2.3. The displacement of the string may be interpreted as the sum of two traveling displacement waves. Without any loss in the string, the two waves merely shift in position as time progression, experiencing no further change in shape or magnitude. This may easily be modeled through the use of digital unit delays, as in Figure 2.18. Figure 2.18 displays a section of a bi-directional DWG, where the upper and lower portions model right-going and left-going traveling waves, respectively. This model represents temporally and spatially sampled waveforms; thus, propagating waves should be band-limited so as to prevent aliasing. Each unit delay represents a section of string of length $X = cT$ where T is the sampling period (the inverse of the sampling rate) and c is the speed of sound in the string. By this notation, the waves travel X meters in T seconds. To recover the displacement of the string at any point, corresponding spatial samples of the traveling waves may be added together.

One may account for the boundaries of a string or other modeled object by considering how waves are reflected. In the case of the rigidly terminated string, waves are reflected without loss, undergoing a sign inversion. Figure 2.19 shows a simple model of a rigidly terminated string. Where there are losses in reflected waves, such as in the

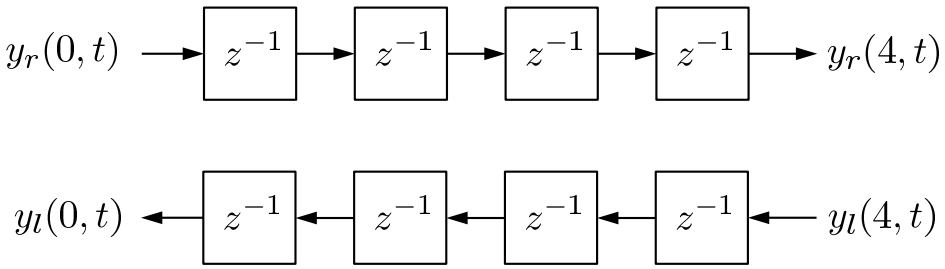


Figure 2.18: A bi-directional DWG section, modeling right-going and left-going propagating waves, without loss. After [2].

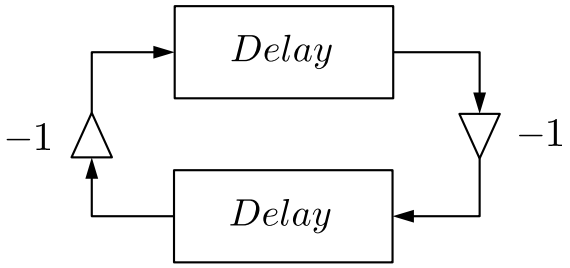


Figure 2.19: A simple bi-directional DWG model, simulating rigid boundaries. Note that excitation and observation are not explicitly shown in this figure. After [2].

bridge of a guitar, one may replace the inverting gains by a filter, representing the losses of reflected waves.

To model uniform losses within the string, as in (2.11), a multiplicative gain, $0 < g < 1$, may be introduced between the delays in a DWG. This is shown in Figure 2.20 for a section of a DWG. Such an implementation of distributed losses can require a great number of multiplications, possibly limiting real-time synthesis. Where calculation of the displacement of the string at all points is not important, however, the delays and gains may be commuted and lumped, as in Figure 2.21, to allow a more efficient (and in

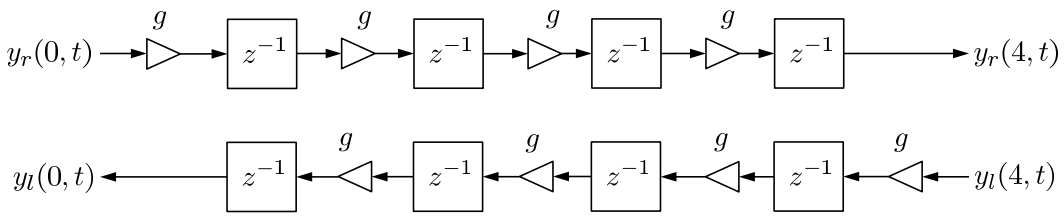


Figure 2.20: A bi-directional DWG section, accounting for frequency-independent losses of the traveling waves. After [2].

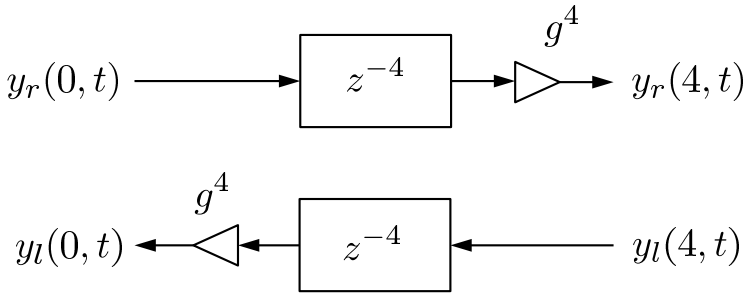


Figure 2.21: A bi-directional DWG section in which distributed losses and delay have been commuted and lumped. After [2].

this case, still exact) realization. This may be done, for example, where the string is only observed at a point, as in the case of an electric guitar's pickup. If propagation losses are frequency dependent, they may be modeled by filters distributed between the delays of the DWG; these may be lumped, however, under the same conditions as frequency-independent losses. Exploitation of lumping and commuting linear elements is perhaps one of the most important and defining characteristics of efficient physical modeling, enabling real-time implementations [2].

To model stiffness in DWG models, one may replace the delays with properly designed allpass filters. A pure integer delay is a special case of an allpass filter with linear phase, whereas other allpass filters may exhibit strong nonlinearity in their phase

responses. Typically, as in the KS algorithm, dispersion due to stiffness is more efficiently modeled by one or a few allpass filters [2,107]. The phase response should be chosen so as to properly model the frequency-dependence of propagating waves in the medium. Note that for very stiff structures, such as xylophone bars, allpass modeling of stiffness may become prohibitive for real-time synthesis [6].

An example DWG model with possible feedback-based interaction at a point and output taken as the wave traveling toward the bridge is shown in Figure 2.22. Similar to models in [2, 3] and elsewhere, the effect of dispersion has been lumped using a single filter on the "nut" side. Justification for placement on this side stems from the prevalence of string instruments being excited near the bridge, resulting in a greater length of string (and hence, more dispersion) on the nut side. Further, in this model, losses are lumped in the bridge filter and nut filter, which may account for losses attributable to both the string and terminations. In the interest of efficiency, model implementations are often further simplified. Such a simplification appears in Figure 2.23, where the nut has been assumed to be perfectly rigid and delay elements have been lumped on each side of the interaction. Note that the output here is not delayed by the bow-to-bridge delay. Where such delay is lossless, this merely provides an advanced output waveform. Even when this propagated wave is attenuated, losses are often small [4].

In this work and elsewhere, DWG models often simulate only transverse vibrations in one dimension. [2] The additional traveling waves in media may be simulated in DWG models by parallel delay lines for each wave type, with proper coupling at terminations and any points at which the different waves interact. An additional dimension of transverse vibration may be simulated for added realism, especially in the case of string terminations that result in variable propagation length in the different planes of vibration [4, 108]. Torsional waves may also be modeled, though their inclusion may have

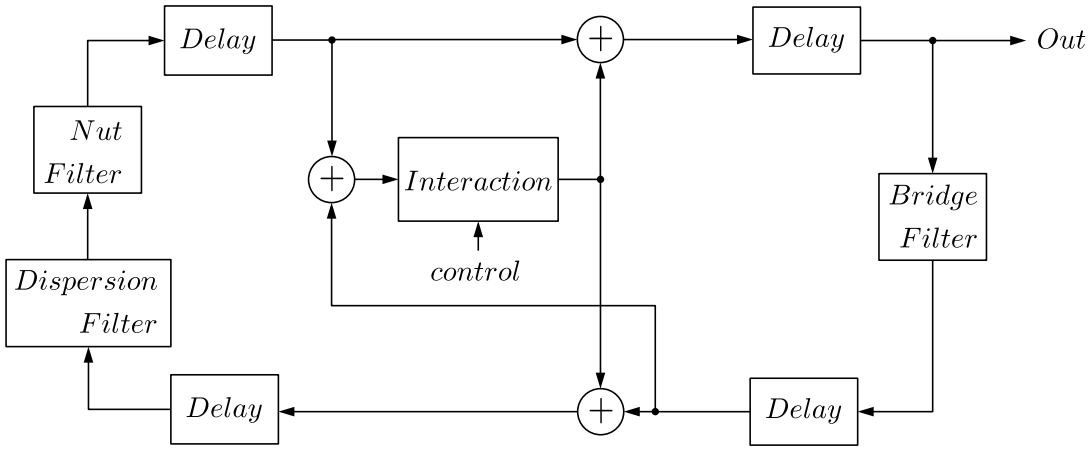


Figure 2.22: A DWG model showing lumped losses and dispersion, allowing for feedback interactions, such as bowing. (After models in [2,3].)

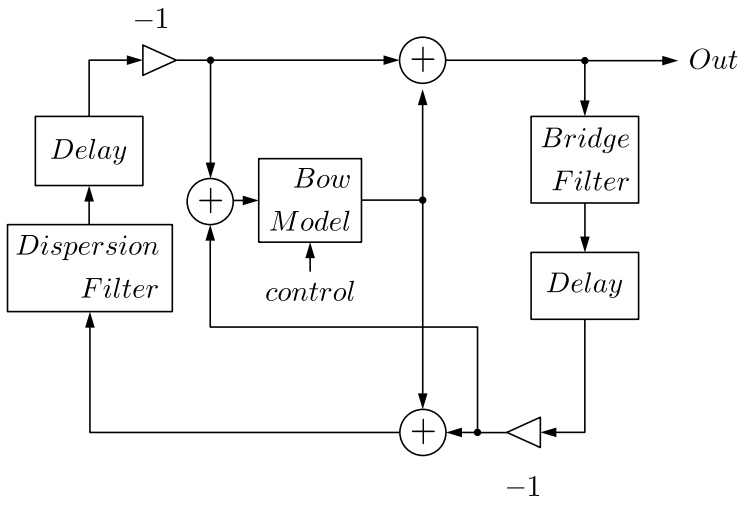


Figure 2.23: A DWG model with further lumping of elements and simplification of the reflection at the nut. (After models in [2,3].)

little effect, as shown for bowed strings [109]. Further, longitudinal wave simulation may benefit sound synthesis, as in the case of [110, 111]. (Note, however, that [110, 111] do not use a DWG model for longitudinal wave simulation.)

Extensions of DWGs include the multi-dimensional DWG mesh [112, 113]. In the two-dimensional mesh, junctions representing sampled portions of some membrane are interconnected by unit delays, accounting for waves propagating away from and into the junction. This may be extended into three dimensions and beyond [112].

Example Application

An example application of a DWG model, illustrating a model of a bowed guitar string without stiffness, is shown in Figures 2.24 to 2.28, with simplifications made for efficiency. Each of these figures shows, via a photograph, the portion of the guitar that is modeled by the darkened part of the model in each respective figure. The output of this model is taken as the wave traveling into the bridge, and may be sent to a body model for further realism. As with many other models here and in [2, 3], many simplifications are made, such as lumping of all losses at the bridge and no loss at the nut.

Singe Delay-Loop Models

Similar to the KS algorithm, single delay-loop (SDL) models are used by some as an efficient phsyically-inspired model for sound synthesis [4, 36, 114, 115]. Derived through manipulation and approximation of a DWG model, SDL models isolate various aspects of a simple physical model relevant to interaction, such as pluck position and excitation, as with KS models [4]. A simple example SDL model is shown in Figure 2.29, as in [4], where the model is separated into an excitation block, a pluck position filter, a string loop, and an integrator (to account for the effect of the bridge). The implementation of SDL models is varied, but common to all forms, as in KS models, is the string loop, which accounts for the modal frequencies and decay rates of a model.

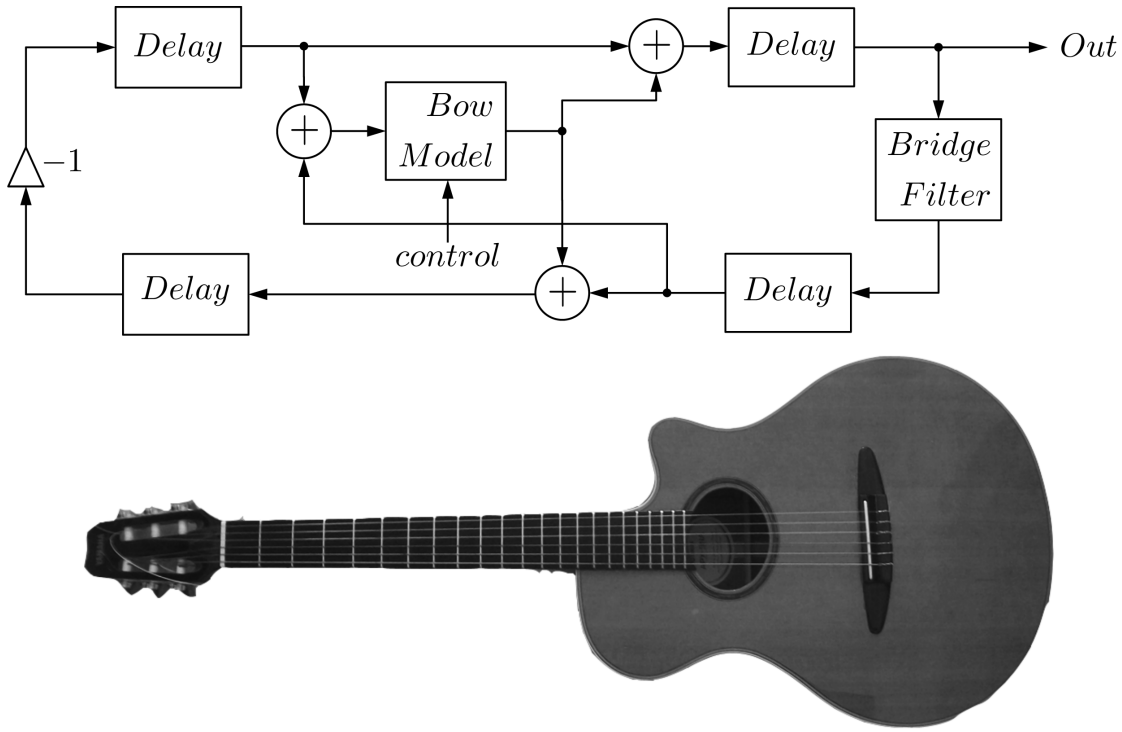


Figure 2.24: An example DWG model representing a bowed guitar string with no stiffness. The nut in this example is modeled as a perfectly rigid boundary, and any losses in the string are lumped at the bridge. Also, the output is taken as the wave traveling into the bridge. The output of this model may be further fed into a model for the guitar body.

2.4.4 Interacting with Physical Models

Many physical models take the form of a source/filter or exciter/resonator model. Such models generally separate an object (the filter or resonator) from its source of excitation (i.e., interactive events with the object). This is a common approach in many areas. In voice, it separates the vocal tract from the pulses of the glottis [14]. In general, it separates whatever excitation instantiates a sound from the excited object. Interactions with physical models may not always be as simple as a one-dimensional signal fed into a filter, but many interactions can be simplified or approximated as such, or, at the very

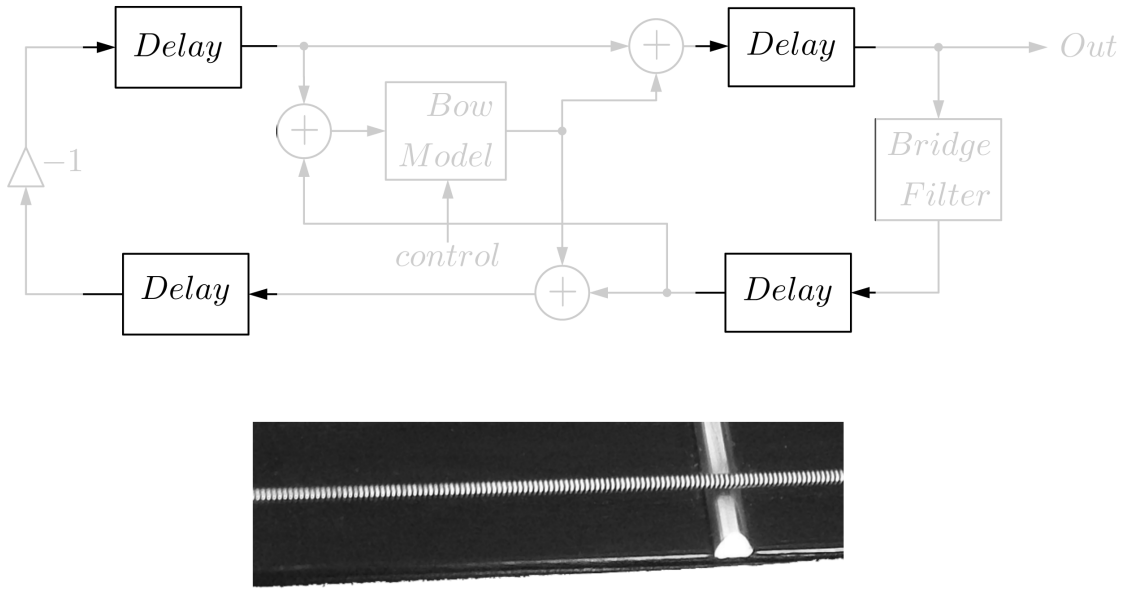


Figure 2.25: The portion of the model that represents the string. Specifically, the left and right-going waves are modeled by the delays.

least, inputs that allow identity resynthesis may be found [91]. Regardless of form, the separation of source and filter is a critical concept in this work.

If a model of an object directly models the physical state and properties of the object, then simulation of interactions requires physical modeling of the manner in which the object is excited. Consider the plucking of a DWG model of a string without stiffness, for example. This may be modeled by initially displacing the string model according the shape of the displacement due to the plectrum before its release. In the DWG model this may be done by loading the delay units with the value of displacement, evenly split among the right-going and left-going portions (assuming no initial velocity) [2]. Consideration of alternative wave variables here and in other interactions can help to simplify interaction [2]. If interaction may be reduced to addition of displacement, velocity, accel-

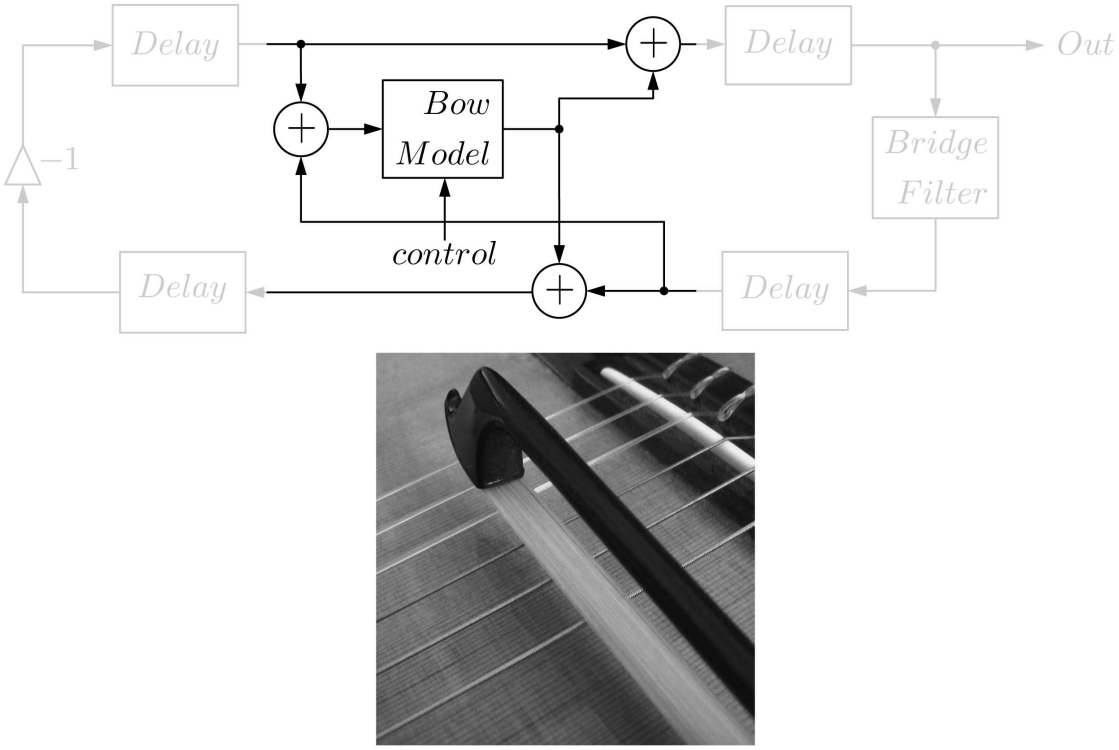


Figure 2.26: The portion of the model that represents the interaction of the bow and string. The bow model may be implemented via any of the methods described here and elsewhere.

eration, or an alternative wave variable at a point, a summing junction may be introduced between delays to handle any input.

Many complex interactions, such as bowing a string, can justifiably be approximated by point-interactions, but with the introduction of nonlinear feedback. How a bow moves the string of a violin, for example, depends on the position and velocity of the bow in a very nonlinear manner [63, 64]. This may be accounted for by observing the wave components traveling into the portion of string that meets the bow and determining the interaction between the bow and string [67]. Examples of DWG bowed string models are given in Figures 2.22 and 2.23, if the interaction is assumed to be an appropri-

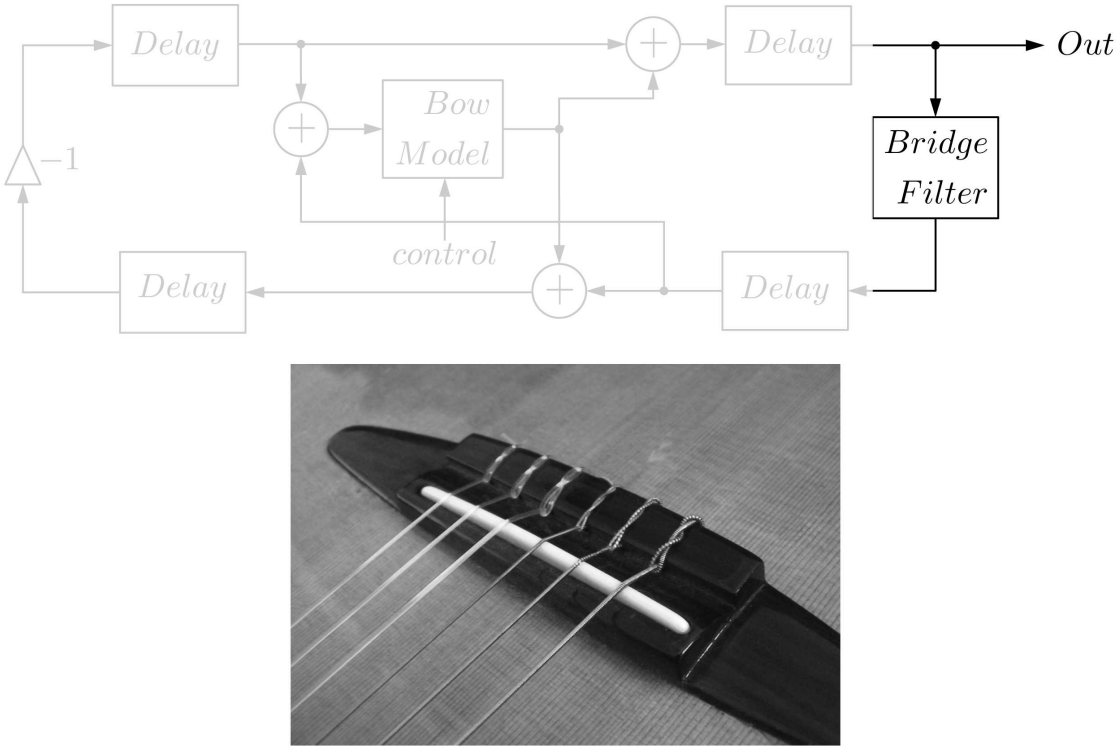


Figure 2.27: The portion of the model that represents the bridge. For efficiency, losses that occur along the string or even at the nut may be incorporated here.

ate bow model, with many of the DWG elements lumped, and the output taken as the disturbances propagating into the bridge. In-depth treatment of modeling friction-based interactions may be found in [3] along with more detailed model structures, particularly those modeling more than unidimensional transverse vibrations.

2.5 Hybrid Physical-Spectral Models

Here, "hybrid" models of physical and spectral models are presented. Many spectral models are particularly well suited for creating slowly changing resonant sounds, but unlike many physical models, they are often insufficient at creating transient sounds [116]. Further, simulation of varying interactions, especially those that are highly state-

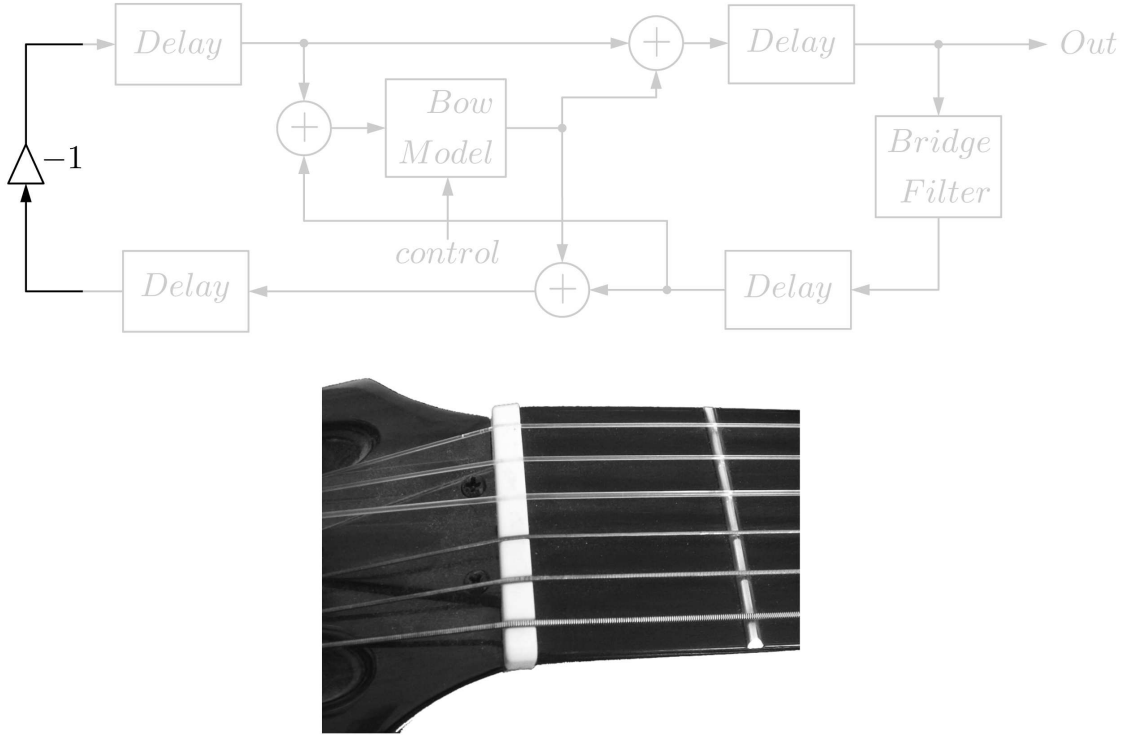


Figure 2.28: The portion of the model that represents the nut. Here, the nut is modeled as a perfectly rigid termination that reflects waves without loss.

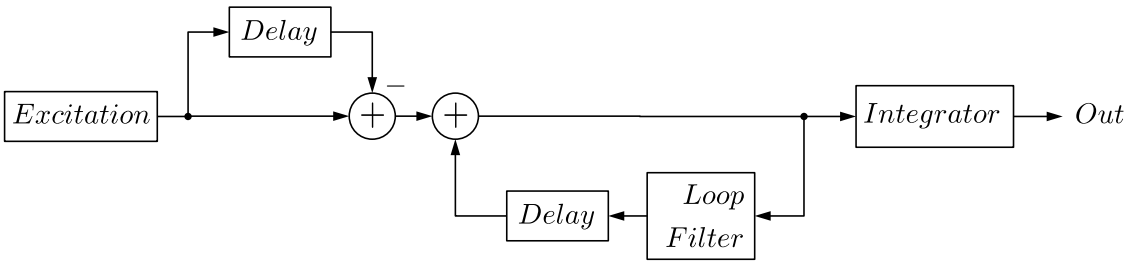


Figure 2.29: A simple single delay-loop model, as in [4] and elsewhere.

dependent such as bowing, is not straightforward in spectral models, motivating hybrid approaches [7]. Such hybrid approaches serve as a "missing link" [31] between spectral and physical models, providing advantages of each modeling approach. Modal models and banded waveguide models are two such hybrid models, modeling physical phenomena in sounding objects, explicitly parameterized by frequency.

2.5.1 Modal Models

Modal synthesis models account for the modal decomposition of vibrations discussed in Section 2.2.4. Models may be classified as distributed or lumped, where distributed modal models account for the mode shapes of an object and lumped generally do not. Distributed modal models for sound synthesis are implemented in [31]. Where mode shapes are unknown, lumped models may just consider the mode frequencies and decay rates. Lumped models may also be implemented for computational ease, or if only the vibration of an object at a point is of interest, allowing mode shape consideration at that point [7]. Further examples of modal synthesis may be found in [117–119]

Lumped modal models are particularly appealing due to the ease of analysis and synthesis. Numerous techniques exist to extract parameters of decaying sinusoids in noise [120, 121]. These may even be as simple as fitting an exponential function to successive amplitude values of peaks found in a time-frequency representation. Synthesis is straightforward using parallel second-order digital resonators, as in [21,118], or by adding together exponentially decaying sinusoidal functions. The use of digital resonators is particularly well suited for modeling linear excitations by an exciter/resonator approach, but can be problematic for implementing complex interactions, such as bowing [6]. An example filter-based implementation of a lumped modal model is displayed in Figure 2.30, where the resonators represent second-order digital filters, each corresponding to a mode of vibration.

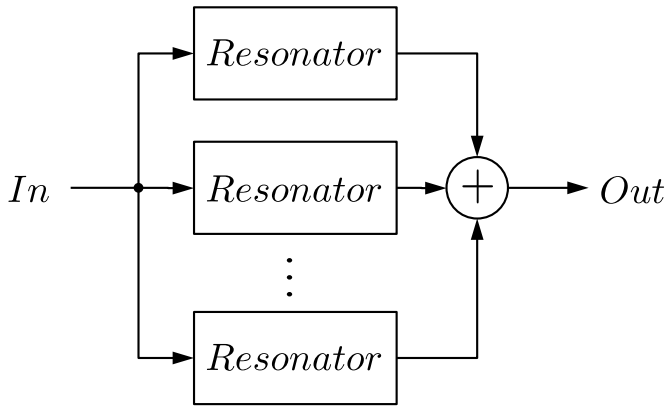


Figure 2.30: A LTI filter implementation of a lumped modal model, with parallel second-order digital resonators, each representing a mode of vibration.

In the filter-based implementation of lumped modal models, the pole angles of the resonators correspond to the modal frequencies, as they determine the frequency of oscillation [122,123]. The radius of the poles, equivalent to the per-sample attenuation of the decaying sinusoids [122, 123], determines modal decay rate. Phase and amplitude of the modes may be used to represent aspects of excitation and position where known [7].

2.5.2 Banded Waveguide Models

Banded waveguide (BWG) models are another type of hybrid model, modeling closed wave trains [70]. BWG models were first introduced in [5] for efficient modeling of bowed bars. The stiffness of bars and other objects requires the design and use of high-order filters for dispersion modeling in DWG models, limiting their use for real-time synthesis, especially when models are dynamically varied. The alternative approach of BWGs separately models each resonant mode by a band-limited propagation model; i.e., separate, band-limited DWGs are used to model the individual resonant modes. A simple BWG structure, after [5], appears in Figure 2.31, where the blocks labeled "BPF" represent bandpass filters, and the "interaction" block may represent any point-interaction

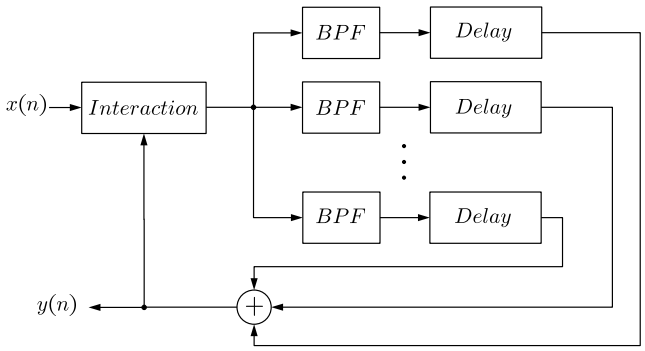


Figure 2.31: A simple BWG structure, as given in [5].

that may rely on feedback. Various implementational structures of BWGs have been used [3, 124], some with slight changes in topology and BPFs [125] or explicit modeling of bi-directional propagating waves [6].

The design of banded waveguides is straightforward from the modal data of an object [6]. Delay lengths are set to the round-trip propagation time of the individual modal frequencies considered; this is dependent on the propagation path length and the speed of sound at the frequency of interest. The bandpass filters are centered about the resonant or modal frequency of each banded waveguide. (Details of this are further addressed in Chapter 4 and Appendix A.) Most implementations [6, 72] use second-order constant-gain digital resonators [30,126], though other choices have been put forth [8,9,125]. The bandpass filter's gain at the resonant frequency may be set to a value that represents the lumped round-trip losses of a propagating wave at the mode frequency; this accounts for the modal decay rate [6,7]. The bandwidth of the bandpass filter is chosen to reject other

resonant frequencies of the modeled feedback path [6,7], but no optimal choice has been put forward, motivating much of the work herein.

Used to model stiff one-dimensional objects, each BWG models the same propagation path, but with varying spatial sampling. A particular strength of BWGs, however, is the ability to efficiently model closed wave trains in objects of higher dimension [6,7]. Beyond models of bars, BWGs have been used to model cymbals, tabla, prayer bowls, glass harmonicas, and other objects [72]. As BWGs model traveling waves, they easily afford simulating interactions such as bowing [5–7].

Chapter 3

RE-SONIFICATION OF ENVIRONMENTS

This chapter examines re-sonification on the scale of environments. To synthesize the sound of an environment requires knowledge of the type of sounds that occur in a place. This may be had from information ranging from a comprehensive description of objects in some space to sound recordings from a locale. Given the diversity of sounds and their origins in different environments, high-fidelity modeling of objects and events within an environment proves difficult in general. Therefore, this work approaches environmental re-sonification through broader soundscape synthesis that plays back recordings made *in situ*.

Previous efforts in soundscape synthesis are varied [16, 17, 127–130]. Some methods rely on identifying and parameterizing sounds within recordings [127, 128]. Other methods, as in this work, rely on identifying relevant and meaningful sound recordings to include in a generated soundscape [16, 17, 129, 130]. This identification process is often controlled by a composer or designer, using prior knowledge [16] or information gathered, such as through interviews [17]. To aid in soundscape design for re-sonifying environments, a method for automated design, using playback of geo-tagged sound recordings is presented. This method, introduced in [12], uses an ontology that relates sounds using acoustic, semantic, and social information [18].

3.1 Re-Sonification of Geographic Activity

Meaningful re-sonification of activity in a geographic region can be difficult when recorded sounds from that region are either 1) abundant or 2) scarce. Where recordings in a region are few in number, re-sonification itself may be sparse or highly repetitive without the inclusion of relevant sounds from other locations. Conversely, if recordings from a region are plentiful, many sounds may be redundant or uninformative about the

area's activity. Both situations may be addressed by classifying and using those sounds that are relevant and important to an area. Traditional classification of sounds within a soundscape (keynote, signal, and soundmark) is primarily focused on their perceptual role to listeners [47, 131]. This classification is area-specific, depending on the perception of sounds as dictated by meaning and prevalence in a community. While the identification of important sounds to an area does not provide this classification, it is able to distinguish which sounds convey the relevant activity of a region, a relevance perhaps best determined by that region's own community.

The concept of community-defined importance of sounds has long been held in the auditory field; in [47], Schafer states,

> Acoustic design should never become design control from above. It is rather a matter of the retrieval of a *significant aural culture*, and that is a task for everyone.

This idea also extends beyond the auditory domain; Google's PageRank technology, for example, determines the importance of web pages by considering the number and relative importance of other pages that link to them [132]. The relevance of such pages is then defined by the internet community's own activity. Similarly, the acoustic knowledge and the actions of a community can help to reveal important sounds for the re-sonification of geographic activity.

To work towards revealing this importance, we use an ontological framework to link sounds together through acoustic, semantic, and social information [18, 133]. Using acoustic content in conjunction with user-provided tags, the framework relies on the prior knowledge of acoustic and semantic ontologies combined with community-defined social links between sounds and concepts. This ontological framework is then used to define the edges of a graph-based generative soundscape model used to re-sonify specified lo-

cations through the playback of sounds in a database. Similar to the use of textual queries to filter a ranked list of important websites, we use location to determine the soundscape model parameters such that geographically relevant sounds play frequently. Consideration of the size (surface area) of locations allows this method of re-sonification to scale to communities or regions of varying size. Using sounds recorded from these locations and elsewhere that are deemed important by an area's community, our methodology aims to create meaningful soundscapes reflective of the geographic sound activity in those areas.

3.2 Ontological Framework

To automatically compose soundscapes from collections of sounds with user-provided descriptions, some notion of similarity between sounds is necessary to determine what sounds may be relevant to a space. For example, if few sounds are recorded in a location, retrieving perceptually similar sounds provides greater diversity in the synthesis process. We calculate such similarity with an ontological framework that links together sounds and concepts, using acoustic similarity between sounds, social information in the form of links between sounds and concepts, and semantic information in the form of conceptual similarity. The ontology effectively finds the shortest distance between sounds, relating them through the similarity of their acoustic features and user-provided tags [18, 133].

To more compactly describe the relation of sounds to one another, we consider a multi-dimensional scaling (MDS) of the sounds [134]. An MDS places the sounds in a lower dimensional space, such that the relative distances between them may be best retained. An example MDS is shown in Figure 3.1, where the dots represent sounds shown with user-provided tags. Placing the sounds into a two-dimensional space serves as a starting point for the automated soundscape design process, described below.

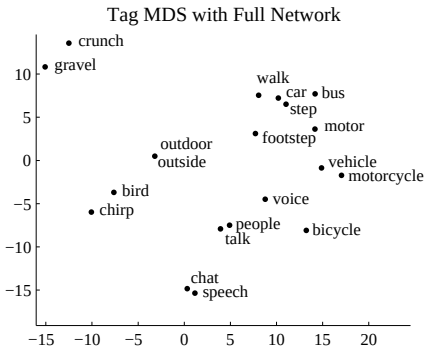


Figure 3.1: An example MDS of tag-labeled sounds.

3.3 Markov Transition Networks

To generate soundscapes for our application, we have chosen to use an emerging compositional structure that we call a Markov Transition Network (MTN), a variation of the models introduced in [15, 16, 135]. An MTN is a directed graph with N nodes, with possible directed edges from each node, i, to another node, j, including $j = i$. An *actant process*, $A(t)$, traverses nodes, transitioning via these edges, taking on values from 1 to N, representing the node at which the process is located at a given point in time. Each edge has an associated transition time, $\Delta(i, j)$. When $A(t)$ "enters" node i, the choice of the next node, j, is determined by an associated probability, $P(i, j)$, and the actant process waits a time of $\Delta(i, j)$ before making the transition. If no edge exists between any two nodes, the associated probability is zero. Given these properties, $A(t)$ is not strictly a Markov process, as transition times are deterministic, depending on the origin and destination nodes. Figure 3.2 displays an example MTN, with nodes and transition times of edges labeled. (Edge probabilities are omitted for clarity.)

Sound synthesis is performed by the sequenced playback of sounds as determined by the actant process. Sounds in the database are uniquely associated with a node, i, and a duration, $D(i)$. Upon $A(t)$ reaching a new node, the associated sound is played back in

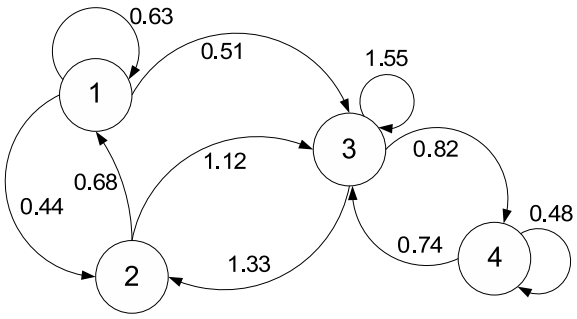


Figure 3.2: Example MTN for soundscape synthesis. Edges are labeled with transition times. Transition probabilities are not shown.

full, regardless of the chosen transition time to the next node or length of the following sounds. We presently mix together all sounds being played back into a single soundscape, though we note that a more complex multi-channel scheme could be adopted, and various effects (e.g., reverb) may be applied to individual sounds or the entire mix. Note that multiple actant processes may be active at any time, independently triggering sounds.

Using an MTN, the sequencing of sounds is made random, but it may be limited by the connections made between nodes. If only a single edge is directed from a node, then the sequencing upon the actant process's selection of that node will be temporarily deterministic. However, if all nodes in an MTN are fully connected, the behavior of the actant process becomes less predictable (dependent on the transition probability distributions). By limiting the number of edges connecting nodes, the sequencing determined by actant processes may be made variable, yet confined by the parameters of the network. This is considered in [15, 16], where limited connections are made between clusters of nodes to specify the behavior of complex sources of sound as predictable sequences. We recognize this effect of limiting connections, but we also wish to examine the overall expected properties of the synthesized output. Therefore, we consider the expected temporal density of all available sounds.

For a sound i, with an intensity value (this may be any meaningful chosen measure, such as loudness), $V(i)$, we define the expected sum of intensities of any instances of sound i at a given time to be the density, $Density(i)$, given by

$$Density(i) = \frac{D(i)V(i)}{T(i,i)}, \tag{3.1}$$

where $T(i,j)$ is the expected time for the actant process to travel from node i to node j (including indirect paths). If the actant process travels directly to j, the transition time will simply be the delay, $\Delta(i,j)$, else it will be the delay, $\Delta(i,k)$, to an intermediary node, k, and the time taken to then reach j. Therefore,

$$T(i,j) = \sum_{k=1}^{N} P(i,k)\Delta(i,k) + \sum_{k=1,k\neq j}^{N} P(i,k)T(k,j). \tag{3.2}$$

Letting Δ, P, and T be $N \times N$ matrices with with elements $\Delta(i,j)$, $P(i,j)$, and $T(i,j)$, respectively, we may express (3.2) in matrix-vector form as

$$T(:,j) = C + Q_j T(:,j), \tag{3.3}$$

where $T(:,j)$ is the j^{th} column of T, the i^{th} element of the $N \times 1$ vector, C, is

$$C(i) = \sum_{k=1}^{N} P(i,k)\Delta(i,k), \tag{3.4}$$

and Q_j is P with the j^{th} column zeroed out. This gives

$$T(:,j) = (I - Q_j)^{-1}C, \tag{3.5}$$

which may be iterated over j.

While this allows us to analyze the density of sounds in a soundscape, for the purpose of design, we seek the ability to specify network parameters to create a desired density of sounds (this density may be determined by the interaction to which the resulting soundscape is applied). As the available sounds and their properties are fixed,

specifying the density value of sounds fixes the desired diagonal elements of T. This leaves flexibility in determining the MTN parameters, Δ and P, as there are N equations (one for each of the diagonal elements of T) and up to $2N^2$ unknowns. Therefore, we allow P to be chosen by the designer (human or computer). By choosing P, the connecting edges of the network may be defined, and connections between relevant or logically successive sounds may be reinforced with high probability. The desired densities may then be achieved through the necessary values of Δ.

To determine Δ, we first define $F = E\Phi$, where $E \in \mathbb{R}^{N\times N}$, $\Phi \in \mathbb{R}^{N\times N^2}$, and the i^{th} row of E is given by

$$E(i,:) \;=\; e_i + q_i(I - Q_i)^{-1}E_i, \tag{3.6}$$

where Q_i is P with the i^{th} column and row removed, q_i is the i^{th} row of P with $P(i,i)$ removed, e^i is the i^{th} row of the size-N identity matrix, E_i is the identity matrix with the i^{th} row removed, and Φ consists of all zeros except for

$$\Phi(i, i + N*(j-1)) \;=\; P(i,j), \tag{3.7}$$

where i and j are iterated from 1 to N. Finding Δ may then be achieved by solving the quadratic program:

$$\begin{aligned} \text{Minimize} \quad & ||F \cdot \mathbf{vec}(\Delta) - \tau||_2^2 \\ \text{subject to} \quad & \mathbf{vec}(\Delta) \succeq b \end{aligned}$$

where b is a vector of elements greater than or equal to zero, and $\tau \in \mathbb{R}^N$ is the column vector where the i^{th} element is the value of $T(i,i)$ necessary to achieve the desired density of sound i. The inequality constraint is introduced to allow future extensions where a minimum delay time between certain sounds may be desired. We note that the amount of nontrivial elements of Δ is limited by the edges of the network, and that in some

cases the actual set of achieved densities may be the best approximation of densities in a squared error sense.

3.3.1 Automated Model Design

Using information from the ontological framework and the sounds themselves, we have developed a method of automatically designing an MTN to re-sonify the sound activity of a specified “virtual environment" that corresponds to a physical location. Seeking to play the sounds from and relevant to the location, we use our ontological framework to make connections between relevant sounds in the MTN and specify the other parameters such that the expected densities of local sounds are relatively high. By making local sounds dense in the soundscape, they will clearly be heard often, making the available local sounds a key component of the soundscape. As this also implies that the actant process will often travel to local sounds, the creation of edges based on relevancy and importance may aid the actant process in traversing nodes corresponding to sounds relevant to the recorded local sounds. This method is executed as follows.

The edges between vertices are determined by performing a Delaunay triangulation (the dual graph of a Voronoi tessellation) on the sound locations in the previously described two-dimensional MDS. Where a line is drawn between two vertices in the triangulation, edges will be created in both directions; self-connections are not made. The results of Delaunay triangulation on the MDS vary with the placement and clustering of sounds, but it generally connects sounds to those nearby (i.e., sounds deemed relevant by the ontological framework) in the MDS. These connections allow the playback of local sounds to often be preceded and/or succeeded by relevant sounds. The triangulation, however, may make some connections between sounds that are not deemed very similar (by the ontology) to one another, but the inclusion of such connections can help to ensure that actant processes do not always concentrate near certain nodes when the

local sounds are spread in the MDS. Use of Delaunay triangulation also guarantees that every vertex will be connected to at least two other vertices, which can help to prevent repetition.

The desired density of sounds is specified to be inversely related to the distance between the sound's location of recording and the user's virtual location. This relation has been implemented as a Gaussian function, referring to the standard deviation as the "listening radius," which sets the size (in surface area) of the region to be explored. The total density of all sounds may be adjusted (so that soundscapes are not overly sparse or dense), perhaps most usefully to a constant value. As previously described, specification of the densities determines the values of the transition times, but requires transition probabilities to be provided. The probabilities may be set arbitrarily, but the choice of probability distribution will affect the achievable densities of sounds. Currently, we set the probabilities so that they may further "encourage" the actant process to travel to local sounds. We achieve this by setting the transition probabilities between nodes such that the ratios between the probabilities of edges emanating from a node are equal to the ratios of the desired densities of the nodes toward which they are directed. In practice, it has been observed that this distribution scheme typically provides better actual densities than a uniform distribution. As this method of soundscape synthesis only requires a virtual location as input when sounds and their corresponding ontological framework are available, it may be applied to various interactions, static or dynamic.

3.4 Soundwalks

Application of our method for re-sonifying geographic activity was previously realized in the form of "Soundwalks," an interactive social website for sharing and tagging sounds and virtually exploring geographic regions. An interactive map allowed users to navigate in a virtual soundwalk mode, "scrubbing" a virtual token across the map, creat-

ing a virtual soundscape. The soundwalk mode features a variable “listening radius" that may be thought of as the radius of a circle that contains the sounds most expected to be heard. It is effectively the size of the area considered in creating the soundscape. The listening radius may be varied from small to large so as to create soundscapes that range from simulating observable soundscapes at small specified locations to providing sonic summaries of large geographic regions. The soundscape is created from an automatically generated MTN as previously described, using a single actant process. Periodic updates of the network parameters are made to adapt to the user’s movement. The actant process (which is initialized to the sound recorded nearest the virtual location) functions continuously, using the MTN as it is updated. A screenshot of the interactive map (with an open information window) in the virtual soundwalk mode appears in Figure 3.3.

While limited deployment and use prevented large-scale assessment, the application was favorably reviewed as providing a sense of activity to otherwise static maps [12]. The most frequent problem observed was the inclusion of keynote sounds (e.g., the beep of a local light rail car, or the cheer of a stadium’s crowd) in inappropriate areas. Additionally, some sounds were played too frequently. Such issues could be addressed through more controlled design or more automated schema. Despite these problems, the proposed method demonstrates the ability to re-sonify geographic activity through environmental soundscape synthesis, guided by community-provided data.

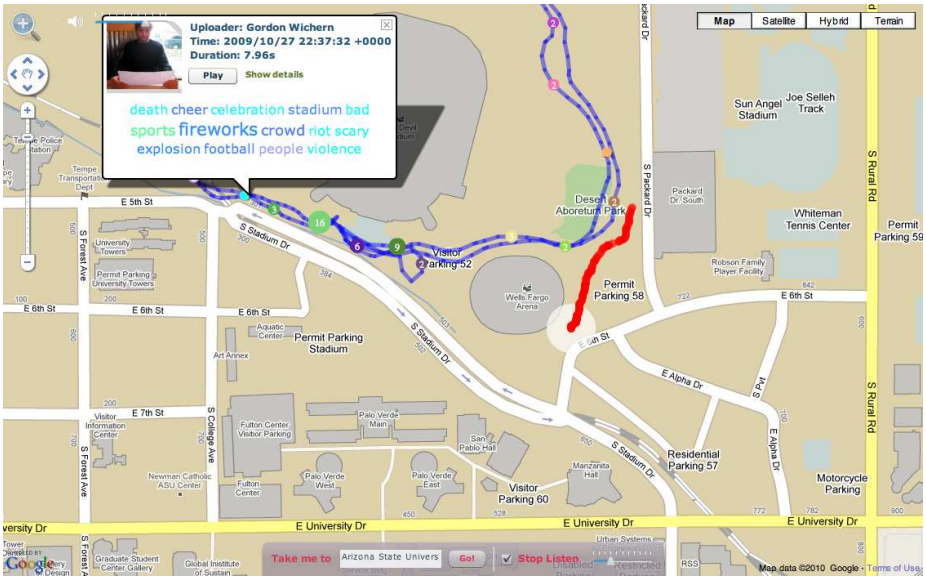


Figure 3.3: A screenshot of the interactive Soundwalks map in the virtual soundwalk mode.

Chapter 4

RE-SONIFICATION OF OBJECTS

This chapter describes the modeling of objects for re-sonification, focusing on objects that exhibit resonant modes as a result of closed wave trains. To simulate sounding objects in an ecologically valid way while permitting a range of transformations and interactions, modeling should consider both physical and perceptual aspects [136]. With no particular application of re-sonification in mind, modeling that arises from recorded sounding objects, prior knowledge, desired parameters, or any combination thereof is of interest. As a hybrid spectral-physical model that easily and efficiently allows modeling of differing resonant objects, banded waveguide (BWG) models are thus considered here for object re-sonification. By modeling wave propagation separately for the modes of a sounding object, independent design and control of each mode is made simple, as compared to DWG models, while maintaining the ability to efficiently simulate complex interactions such as bowing [7]. Explicitly parameterized by both physical and spectral parameters, BWG models easily permit both physical and perceptual transformations of resonant objects [21, 137].

Much of the focus here is on alterations and refinements to BWG models, first presented in [8] and [9], so as to improve the correspondence of simulations with that of modal and DWG models, such that existing and emerging methods of analysis, transformation, and synthesis for DWG and modal models may be appropriately applied in re-sonifying objects with BWG models. General concepts in creating and transforming object models are first discussed in this chapter. This is followed by a description of previous BWG models and two proposed approaches to improving BWG modeling. Lastly, concluding remarks are given.

4.1 Creating and Transforming Object Models

Creation of BWG models from sound recordings may make use of modal analysis methods (e.g., [120, 138]) to find the modal frequency and decay. Amplitude and phase information may also be found, but it is dependent on interaction and observation. Although one may not accurately recover mode shapes or propagation length from single sound recordings (without appropriate prior knowledge or assumptions), BWG models may be created from sound recordings to a describe a virtual object from which a recorded sound *could* have emanated. For example, if an inharmonic sound is analyzed, one might try to fit a one-dimensional model of a stiff object, such as a bar. Alternatively, one might consider the sound to have come from some multi-dimensional object in which the resonances arise from separate closed wave trains in the object. Rather than view this as a deficiency, this may be considered a point of design, allowing modeling in object re-sonification to be highly controllable.

As discussed in Section 2.5.2, the hybrid spectral/physical nature of BWGs affords manipulation of both physical and spectral parameters (and relatedly, perceptual parameters [137]) . Spectral transformations of BWGs involve changing the modal frequencies; this may be carried out by adjusting the the band-limiting filters and propagation delay lengths appropriately. Such a transformation may be used to easily change the partial frequencies or timbre of a sound. The time-duration of a BWG model may be adjusted by modifying the gains so as to increase the decay time of each mode. Alternatively, the interaction with the model may be adjusted to lengthen the sound's duration.

Physical transformations of BWGs may be performed by considering the effect of various physical properties. The relative perceived size of a modeled object may be adjusted by increasing or decreasing the propagation delay lengths (and appropriately re-tuning the band-limiting filters). Adjustment of the gains is not straightforward without

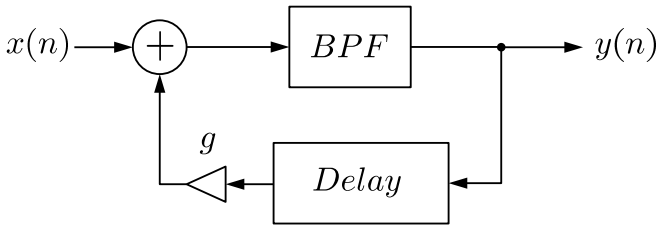


Figure 4.1: A single BWG. (After [3]).

full knowledge of what losses are due to general propagation, reflection, or resonance in some "body." The perceived material of an object may be modified by distribution of the modes and their decay rates, as in [21].

4.2 Previous Implementations of Banded Waveguides

A single BWG, as described in [3, 5–7, 72], with lumped elements, and implemented in a filter-like manner with additive interaction at a point, appears in Figure 4.1. This represents a single delay line (SDL) [4] or KS-type implementation of a BWG subject to interaction and observation at a point. As in [3, 5–7, 72], we consider a constant-gain digital resonator for use as the bandpass filter. The filter's center frequency and the loop gain are determined by the desired or modeled modal parameters, but the bandwidth of the filter has no strict physical interpretation [6]. As shown in [3], the impulse response differs between a modal resonator and a BWG designed to have the same decay and frequency. Examples of this are shown in Figures 4.2 to 4.7, where comparable modal and BWG models of a decaying sinusoid with modal frequency of 441 Hz are shown (simulated with a sampling rate of 44100 Hz), with varying bandwidths of the bandpass filter

Clearly, the impulse response of the BWG can greatly vary, dependent on choice of bandwidth. A small bandwidth is desirable, in that it will attenuate the other resonances due to the feedback path; however, narrowing of the bandwidth extends the decay

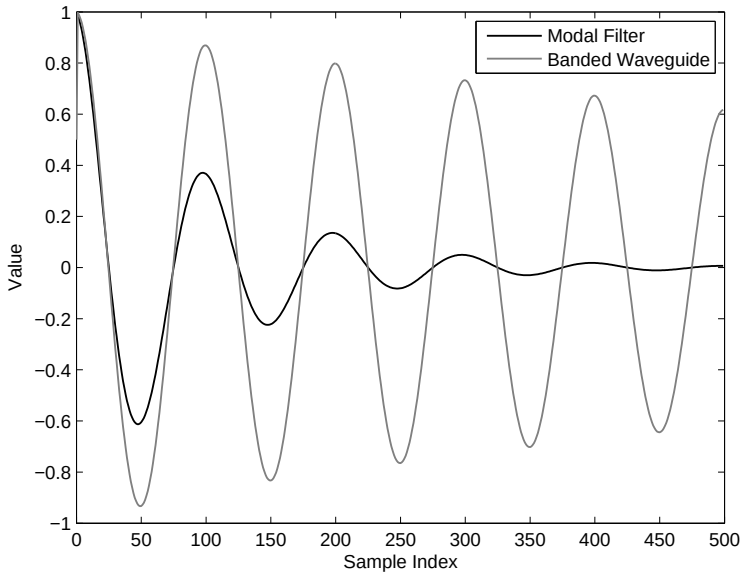


Figure 4.2: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 20 Hz. (Scaled for comparison.)

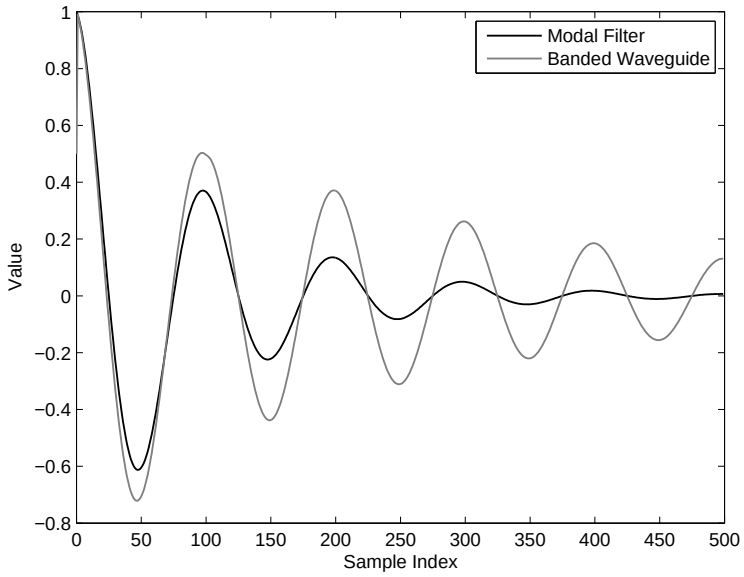


Figure 4.3: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 100 Hz. (Scaled for comparison.)

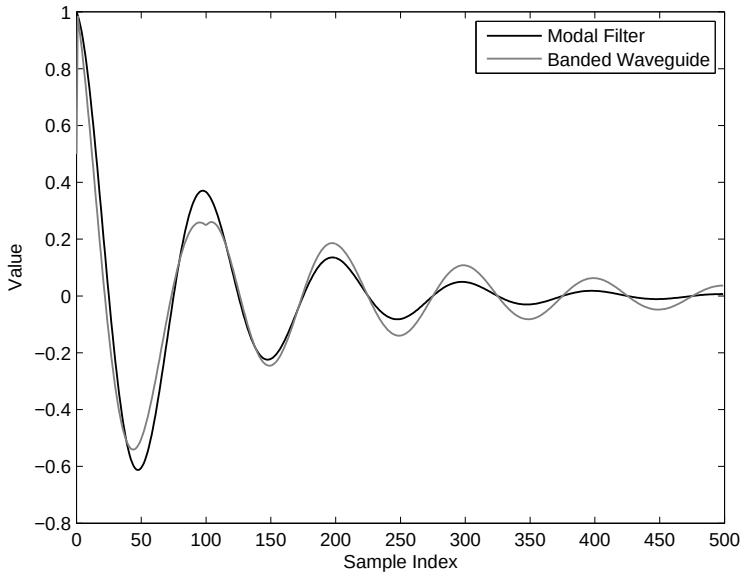


Figure 4.4: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 200 Hz. (Scaled for comparison.)

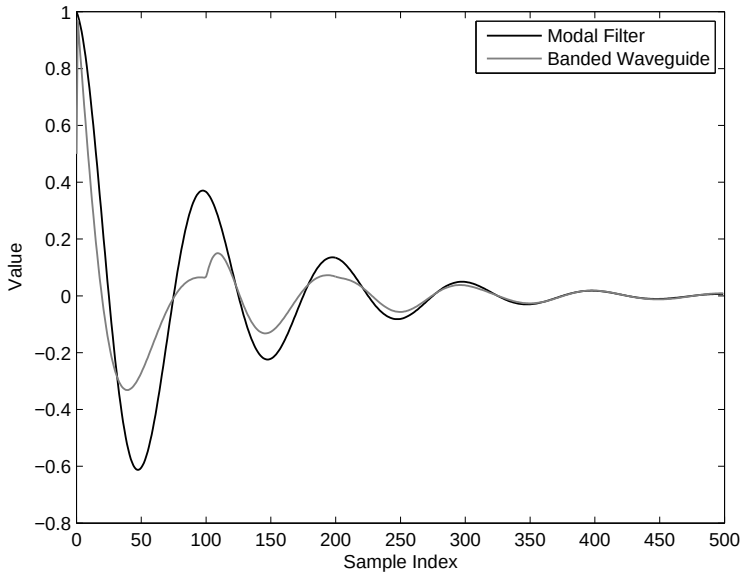


Figure 4.5: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 400 Hz. (Scaled for comparison.)

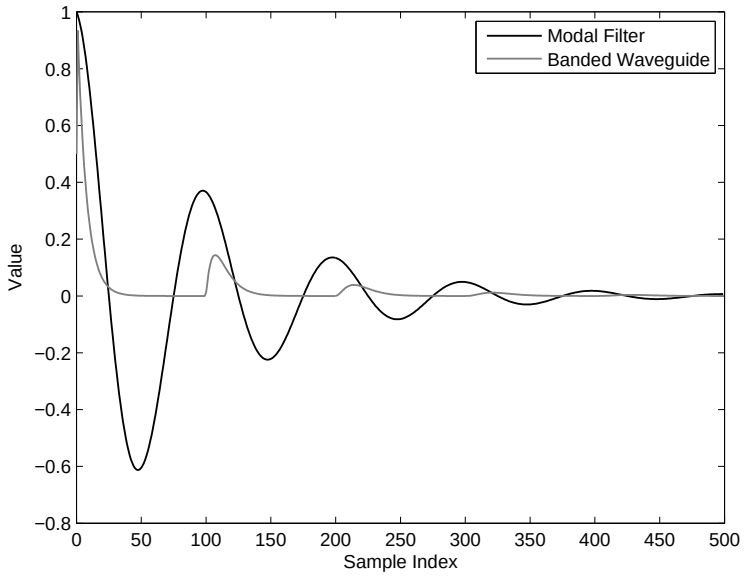


Figure 4.6: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 2000 Hz. (Scaled for comparison.)

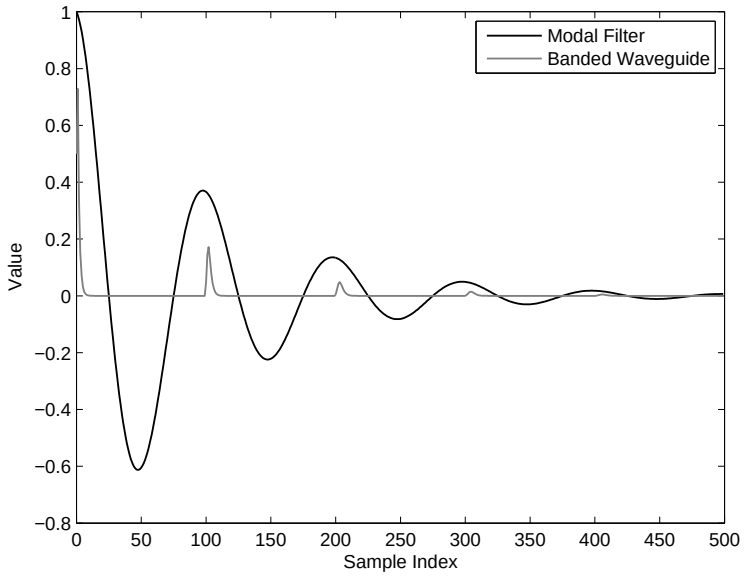


Figure 4.7: A comparison of the response of a modal filter and comparable BWG with a bandwidth of 10000 Hz. (Scaled for comparison.)

time, as seen in Figure 4.2. This is due to the lengthy impulse response of the BPF. If the BPF's impulse response is longer (by some measure) than the round-trip propagation time, the decay time could extend well beyond desired values, even if the round-trip attenuation was reduced to *zero*, with the BWG's output coming entirely from the BPF. Alternatively, a BPF with a wider bandwidth provides the advantage of a more accurate decay time, as in Figure 4.5, but it may not strongly attenuate other resonances, resulting in artifacts in the waveform and spurious resonances, as compared to the response of the modal resonator. More extreme examples of a very wide bandwidth may be seen in Figures 4.6 and 4.7. This difference in output due to varying bandwidths of the BPF has been used for timbre manipulation in [139], but this is not a desirable feature of physical modeling.

Other implementations of BWG models have been put forward. For example, in [125], fourth-order bandpass filters are used outside of the delay loop for each BWG. This implementation, however, still does not produce ideal modal responses. With no ideal implementation of BWGs put forth by others, attempts to improve BWG models are given below. These alternative implementations of BWGs were first presented in [8] and [9].

4.3 Banded Waveguides Using a Perfect Reconstruction Filterbank

To address the imperfect response of BWG models, as compared to comparable models, this section discusses the use of an individually tunable perfect-reconstruction filterbank to implement BPFs in BWGs, first proposed in [8]. The filterbank is designed about a set of select frequencies (the resonant modal frequencies, in this case), such that each filter passes a single select frequency with unity gain and no phase change, while having zero gain at the other select frequencies. When all the filters of the filterbank have the same input signal, the sum of their outputs is identical to the input signal. The follow-

ing describes the application of this filterbank to the modeling of a perfectly-harmonic bowed string with no stiffness, showing improved results with the filterbank. As noted, real-time use and application beyond a simple case such as a perfectly or near-perfectly harmonic string is not straightforward. However, the results show the feasibility of improving BWG models, even subject to nonlinear interactions.

4.3.1 Perfect Reconstruction Filterbank Design

The design method outlined here, first described in [8], relies on a tree-type structure of complementary filters, spectrally warped via suitably designed allpass filters. Spectral transformation of filters may be performed by replacing unit delays with allpass filters. Low-order allpass filters may be used to turn prototype low-pass filters into high-pass filters, band-stop filters, bandpass filters, or low-pass filters with different frequency specifications [140]. Use of higher-order allpass filters for warping can result in any arbitrary amount of notches, resonances, stop-bands, or pass-bands [141, 142]. As allpass-based warping maps a simple traversal of the unit circle in the z-plane to a more complex traversal – or multiple traversals – basic prototype filter properties, such as the magnitude of passband ripple, are preserved in warped regions.

To isolate the various modes of a harmonic BWG model with the desired filterbank, it is required that each bandpass filter passes the corresponding modal frequency, ω_m, with unity magnitude response and no phase shift. Also, each bandpass filter should completely reject the other modal frequencies. That is, each bandpass filter in the filterbank should have notches (with a magnitude response of zero) at the other modal frequencies. To achieve this specification, one may begin with a Haar filter pair as complementary prototype filters. The Haar filter pair is defined by the transfer functions,

$$\begin{aligned} H_{LP}(z^{-1}) &= 0.5 + 0.5z^{-1} \\ H_{HP}(z^{-1}) &= 0.5 - 0.5z^{-1}. \end{aligned} \tag{4.1}$$

Note that each Haar filter passes a certain frequency (0 or π) with no attenuation and phase shift, while the other filter rejects the same frequency. Additionally, the filter pair is perfect reconstruction, meaning that if the same signal is passed through both filters, the sum of their outputs is identical to the original input signal. This property is preserved when the filters are identically warped.

Using a high-order allpass filter to warp the Haar filter pair, one obtains a pair of filters with multiple passbands and notches, each passing frequency content rejected by the other. If the peak magnitude response and notch of a prototype filter are alternatingly mapped to subsequent modal frequencies, and the complementary filter mapped via the same warping filter, one obtains a pair of filters, each of which passes half (or approximately half, depending on the total number of modal frequencies) of the modal frequencies, while rejecting the others. If each filter's output is then passed through a new pair of complementary filters, designed in the same manner for the modal frequencies passed by the respective filter of the first pair, the new filters then each pass and reject half of the remaining frequencies. Carrying out this process recursively, cascading these filter pairs in a tree-like manner until each path results in the passing of a single modal frequency, then results in the desired filterbank.

To carry out this process, at any stage, with M modal frequencies, $\omega_1, \omega_2, \ldots, \omega_M$, present, one must design an allpass warping filter that maps $[0, \omega_1]$ to $[0, \pi]$, $[\omega_1, \omega_2]$ to $[\pi, 2\pi]$, ..., and $[\omega_M, \pi]$ to $[M\pi, (M+1)\pi]$. When each of the prototype filters are thus warped, each will pass half of the select frequencies, while rejecting the others. The eventual resulting filterbank will be composed of a lowpass filter, a highpass filter, and a number of bandpass filters equal to the total number of modes (excepting DC and the Nyquist frequency). This design may be applied to any set of arbitrary mode frequencies, though numerical problems may arise, as detailed below in Section 4.3.1.

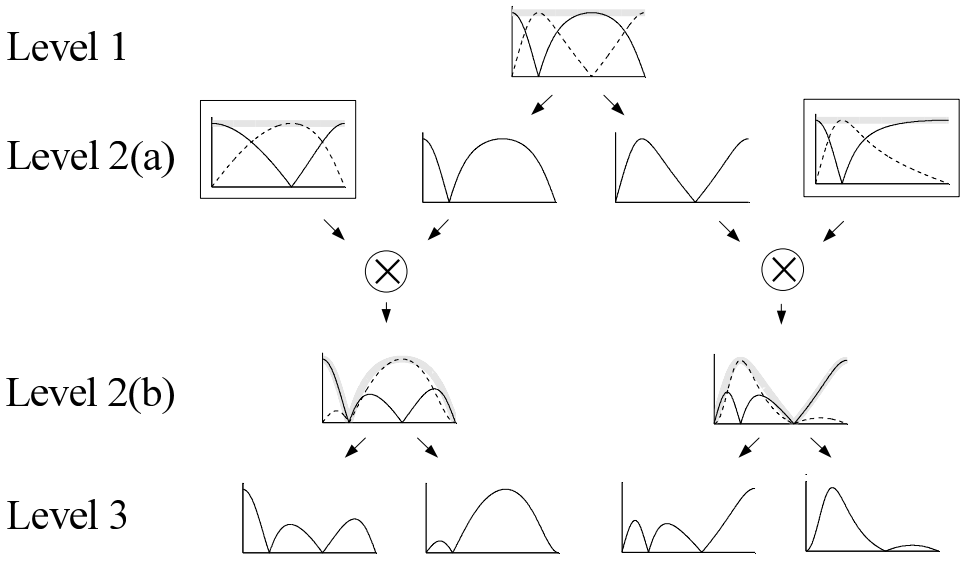


Figure 4.8: A graphical representation of the iterative filterbank design process for the simple case of four modal frequencies, including 0 and π.

It is also important to note that stability and efficiency may generally be improved by implementation using an actual tree-like structure, rather than separate filters for each mode.

Example Filterbank

To illustrate this process, we consider the design of such a filterbank with four modal frequencies (including DC and the Nyquist frequency): $\omega_0 = 0, \omega_1 = 0.2\pi, \omega_2 = 0.6\pi$, and $\omega_3 = \pi$. A graphical representation of the design procedure is shown in Figure 4.8. The resulting filterbank is composed of a lowpass filter, two bandpass filters, and a highpass filter, seen in Figure 4.9 and level three of Figure 4.8. The first step in this example is the design of a suitable allpass filter, $A_1(z^{-1})$, to warp the Haar filter pair such that one filter passes ω_0 and ω_2, while the other filter passes ω_1 and ω_3. This allpass filter must warp the frequency range $[0, \omega_1]$ to the frequency range $[0, \pi]$ of the original filter pair. Similarly, $[\omega_1, \omega_2]$ must be warped to $[\pi, 2\pi]$, and $[\omega_2, \pi]$ to $[2\pi, 3\pi]$

Using an allpass filter with these specifications, the warped lowpass filter will pass frequencies 0 and ω_2 with no attenuation or phase change, while wholly rejecting ω_1 and π. (The magnitude response of the corresponding filter is shown in Figure 4.8 as the solid line plot of level 1 and its leftmost child.) The warped highpass filter will pass spectral content at ω_1 and π while rejecting 0 and ω_2. (The magnitude response of the corresponding filter is shown in Figure 4.8 as the dashed line plot of level 1 and its rightmost child.) Since both prototype filters are perfect reconstruction, this property is preserved when they are warped by the same allpass filter, $A_1(z^{-1})$; the frequency response of the sum of the two filters' outputs is shown by the thick gray line of level 1 in Figure 4.8.

To complete the filterbank design, two more allpass filters must be designed so as to warp filter pairs that further separate the spectral content passed by each filter of the first warped pair. The first allpass filter, denoted $A_{2,1}(z^{-1})$, maps $[0, \omega_2]$ to $[0, \pi]$ and $[\omega_2, \pi]$ to $[\pi, 2\pi]$, such that the warped lowpass filter passes DC and the warped highpass filter passes ω_2. The second allpass filter, denoted, $A_{2,2}(z^{-1})$, maps $[0, \omega_1]$ to $[0, \pi]$ and $[\omega_1, \pi]$ to $[\pi, 2\pi]$, such that the warped lowpass filter passes π and the warped highpass filter passes ω_1. The magnitude frequency responses of these filter pairs are shown in the leftmost and rightmost elements of level 2(a) of Figure 4.8. When these new filter pairs are cascaded with the first filter pair (warped using $A_1(z^{-1})$), the final filters of the filterbank will be obtained:

$$\begin{aligned} H_0(z^{-1}) &= H_{LP}(A_1(z^{-1})) \cdot H_{LP}(A_{2,1}(z^{-1})) \\ H_1(z^{-1}) &= H_{HP}(A_1(z^{-1})) \cdot H_{HP}(A_{2,2}(z^{-1})) \\ H_2(z^{-1}) &= H_{LP}(A_1(z^{-1})) \cdot H_{HP}(A_{2,1}(z^{-1})) \\ H_3(z^{-1}) &= H_{HP}(A_1(z^{-1})) \cdot H_{LP}(A_{2,2}(z^{-1})). \end{aligned} \tag{4.2}$$

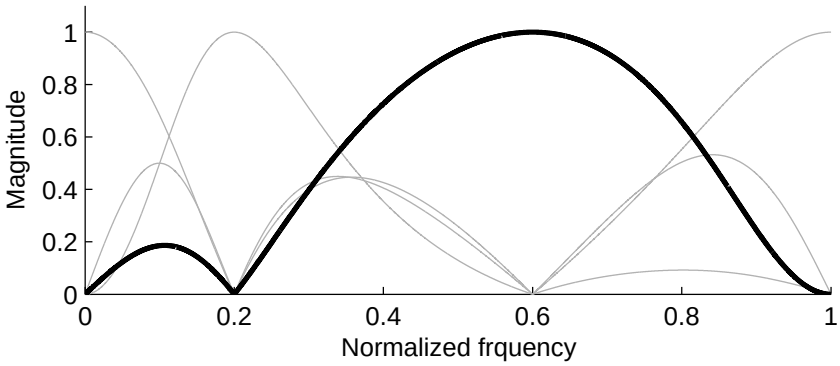


Figure 4.9: Example filterbank. $|H_0(z^{-1})|, |H_1(z^{-1})|, |H_2(z^{-1})|$ (blackened), and $|H_3(z^{-1})|$, with $\omega_1 = 0.2\pi$ and $\omega_2 = 0.6\pi$.

The magnitude frequency response of these filters is shown in Figure 4.9 and level 3 of Figure 4.8. As complementary filters are used at each stage of the design, the whole filterbank retains the property of perfect reconstruction.

For the general case, given M mode frequencies $\omega_1, \omega_2, \ldots, \omega_M$, we obtain an allpass filter that maps the frequency range $[0, \omega_1]$ to $[0, \pi]$, $[\omega_1, \omega_2]$ to $[\pi, 2\pi]$,..., and $[\omega_M, \pi]$ to $[M\pi, (M+1)\pi]$. The warped prototype filters will each pass half the selected frequencies (including 0 and π as selected frequencies) while rejecting the other half. Recursively repeating this process for the remaining frequencies from each of the complementary filter pairs at each level and cascading the properly warped prototype filters then yields a bank of perfect-reconstruction filters. The resulting bank will be made of M bandpass filters, plus single lowpass and highpass filters. This process can be extended to any set of mode frequencies, although numerical problems may arise [143]. In addition, round-off error may also cause the filterbank to become unstable. This is often, though not always, avoided by implementing the filterbank as a series of separate, cascaded filters.

Warping Allpass Filter Design

Here, design of allpass filters to meet the specifications of the filterbank design process is described. This primarily requires allpass filters with an unwrapped phase response that is a multiple of $-\pi$ at specified modal frequencies. (Note that the sign of the phase response is inverted as compared to the warping range frequencies given previously.) This design may be achieved exactly via closed-form equations. We use much of the notation as in [143, 144] in the design procedure.

An N^{th}-order allpass filter,

$$A(z^{-1}) = \frac{a_N + \ldots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}, \tag{4.3}$$

will have a phase response given by

$$\phi(\omega) = -N\omega + 2\arctan\left(\frac{\mathbf{a}^T \mathbf{s}(\omega)}{1 + \mathbf{a}^T \mathbf{c}(\omega)}\right), \tag{4.4}$$

where

$$\begin{aligned} \mathbf{a} &= \begin{bmatrix} a_1 & a_2 & \ldots & a_N \end{bmatrix}^T \\ \mathbf{s}(\omega) &= \begin{bmatrix} \sin(\omega) & \sin(2\omega) & \ldots & \sin(N\omega) \end{bmatrix}^T \\ \mathbf{c}(\omega) &= \begin{bmatrix} \cos(\omega) & \cos(2\omega) & \ldots & \cos(N\omega) \end{bmatrix}^T. \end{aligned} \tag{4.5}$$

To find a suitable allpass filter at any stage, one must then find coefficients such that the phase response at select frequencies matches the desired response. During any stage of the design process, when K ordered modal frequencies, $\omega_1, \omega_2, \ldots, \omega_K$, are being considered, one must use an N^{th}-order allpass warping filter with $N = K + 1$. The allpass filter should have a phase response equal to increasing integer multiples of $-\pi$ at the K modal frequencies.

Denoting the desired phase response at frequency ω_k by $\phi_{des}(\omega_k)$, we define

$$\beta_k = -\frac{1}{2}(\phi_{des}(\omega_k) + N\omega_k), \tag{4.6}$$

such that the coefficient vector, **a**, that satisfies our requirements will be a solution of

$$\mathbf{Fa} = \mathbf{b} \tag{4.7}$$

where

$$\mathbf{F}_{k,n} = \sin(\beta_k + n\omega_k) \quad n = 1,2,\ldots,N \quad k = 1,2,\ldots,K$$
$$\mathbf{b} = -\begin{bmatrix} \sin(\beta_1) & \sin(\beta_2) & \ldots & \sin(\beta_K) \end{bmatrix}^T, \tag{4.8}$$

and $\mathbf{F}_{k,n}$ is the element in the k^{th} row and n^{th} column of **F**. As our desired phase response only requires

$$\phi_{des}(\omega_k) = -k\pi \quad k = 1,2,\ldots,K, \tag{4.9}$$

and $K = N-1$, our system is overdetermined, giving us a set of feasible coefficient vectors. This set should be limited to stable filters. For our design, we choose to further specify the phase response at a single frequency, constraining

$$\phi_{des}(\frac{\omega_1}{2}) = -\frac{\pi}{2}. \tag{4.10}$$

That is, the phase response of the allpass filter at half of the first mode frequency is fixed at $-\frac{\pi}{2}$. This will give a unique solution of (4.7), ensuring the stability of the designed allpass filter, in addition to choosing a solution with a relatively smooth phase response. The phase response of many solutions to (4.7), without the extra constraint given in (4.10), is mostly flat, with small regions of large change due to pole-zero pairs very close to the unit circle. Use of such warping filters results in some transformed filters having very small bandwidth, while others have a large bandwidth (possibly extending beyond adjacent mode frequencies, at which notches appear in the frequency response) and very high stop-band magnitude response. Figure 4.10 shows such an undesirable filterbank. Choosing a smooth response, with less abrupt changes in phase shift, causes more consistent and desired behavior of the filters, as originally shown in Figure 4.9. This constraint is not

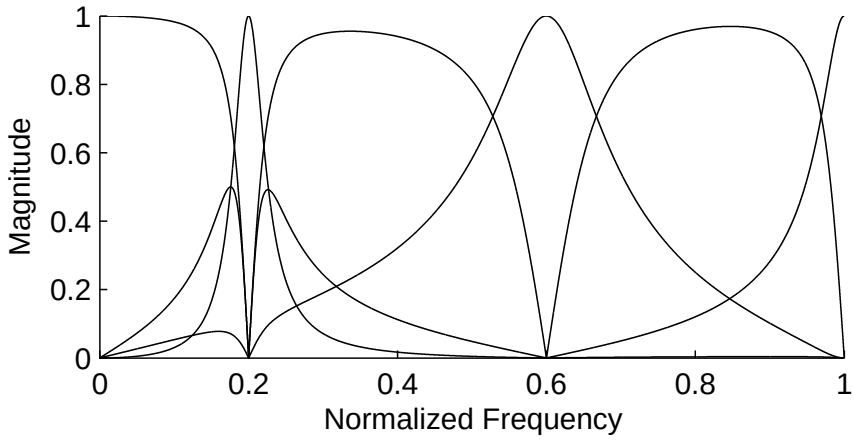


Figure 4.10: Example filterbank ($\omega_1 = 0.2\pi$ and $\omega_2 = 0.6\pi$) with poor response due to the lack of an added constraint.

designed to meet any optimal criterion, but it has worked well in practice to smooth phase responses of the allpass filters. Responses that are more ideal in some sense could be designed through choice of an error function, as in [143].

Implementations of this design method may be prone to numerical problems in solving the linear system of (4.7), particularly in the case of closely-spaced modes, as noted in [143]. This and other sources of round-off error can lead to instability, requiring consideration in design and use.

4.3.2 Simulations

To compare the use of our filterbank with that of biquad filters, we use a simple one-dimensional bi-directional digital waveguide model with losses lumped at the "bridge". We use a bowing interaction, following the friction model of [67] with velocity output. The waveguide model used is that of a generic string with no stiffness and a fundamental frequency of 441 Hz. In this case, "banding" the waveguide is not necessary to physically model the sounding object, but we wish to compare the fidelity of the perfect reconstruction model against the biquad banded waveguide in reference to a

(a) Regular waveguide.

(b) Banded waveguide using biquad filters.

(c) Banded waveguide using the perfect reconstruction filterbank.

Figure 4.11: Simulation results for a 441 Hz bowed string, including detail of steady-state oscillations.

standard digital waveguide, which is made possible by these test conditions. The biquad filters were set to have bandwidths of 44.1 Hz (an arbitrary choice). Each of the BWG models assumes perfectly harmonic partials, and all simulations here use a sampling rate of 44.1 kHz.

Figure 4.11 displays coarse aspects of the transient response (the amplitude envelope over the first three seconds) as well as details of the steady state oscillations (with the regular waveguide oscillation overlaid in gray for the other models) of all three simulations. We can see that the transient response characteristics of the perfect reconstruction model are significantly improved over those of the biquad model; however, there is still

room for improvement when compared against the regular waveguide model. Steady-state response characteristics are perceptually similar among all three models. The steady-state waveform of the perfect reconstruction model, however, follows almost sample-for-sample that of the standard waveguide model.

The above results demonstrate that BWG implementations can be improved, even for nonlinear interactions, but the preceding example does not readily extend to cases beyond simple harmonic objects. In the above design, the filterbank's center frequencies are intended to be the resonant modal frequencies of the entire model. Given a harmonic object, the modal frequencies are the same as the undesirable resonant frequencies of each individual BWG's feedback path. For an inharmonic object, however, this is not the case. Also, the filterbank is of a rather large order (each "level" is of the order of the number of modes), in opposition to the desire for computational efficiency. Therefore, alternative filters for improving BWG implementations are presented below.

4.4 Digital Waveguide-Derived Implementation of Banded Waveguides

Here, we consider an alternative implementation of BWGs, derived by making alterations and approximations to KS and DWG models. The models discussed here were first published in [9].

4.4.1 Derivation

For simplicity in describing the derivation, consider the string loop of single delay-loop (SDL) model [36], effectively a KS-type model, as given in Figure 4.12. Here, the losses, dispersion, and propagation delay of a corresponding bi-directional DWG model have been commuted into a set of filters in the SDL's feedback loop. In Figure 4.12, z^{-L}, $F(z)$, $D(z)$, and $G(z)$ represent the integer delay, the fractional (tuning) delay, a dispersion filter, and a lossy loop filter, respectively. Physically, this corresponds to

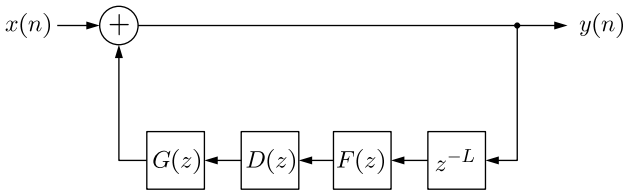


Figure 4.12: A single-delay loop digital waveguide model.

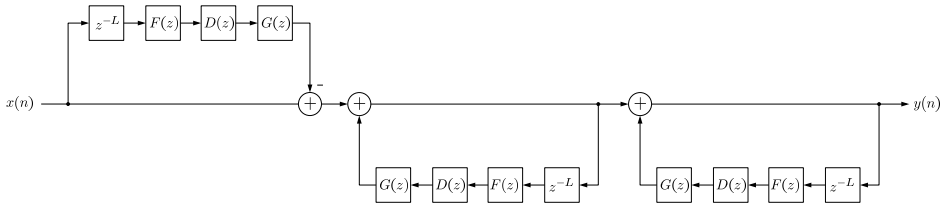


Figure 4.13: A functionally identical single-delay loop model.

linear interaction and observation of a unidirectional wave at a single point. (Choice of wave variables is unimportant in the derivation.)

To derive the model, first consider filtering the input with an identical string loop and the inverse feedforward system, as shown in Figure 4.13, such that the input to and output of the original string loop remains identical. Assuming the direct-form transfer function of the SDL model has no repeated poles and is proper (typical SDL models will be proper, due to the somewhat large integer delay), it may equivalently be represented by its partial fraction expansion (PFE),

$$S(z) = R_0(z) + R_1(z) + \ldots + R_{M-1}(z) \tag{4.11}$$

where each $R_m(z)$ is a resonant filter, representing a mode of the system. Most of the terms are second-order IIR filters, though first-order modes at DC and the Nyquist frequency are typical. Using this modal expansion, the model of Figure 4.13 may be identically restructured into that of Figure 4.14.

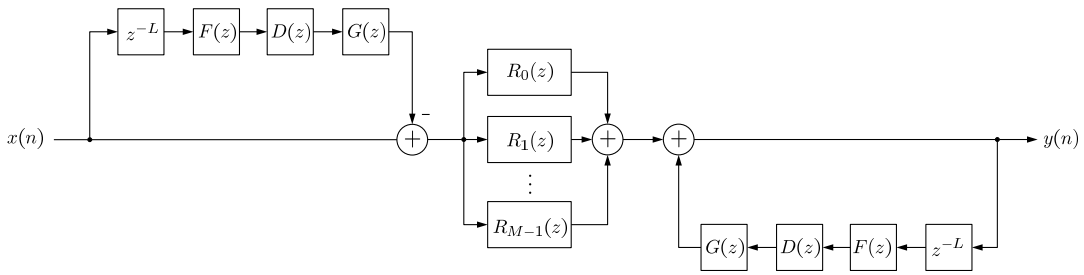


Figure 4.14: A functionally identical, expanded single-delay loop model.

By linear system theory, the feedforward and feedback loops of Figure 4.14 may distributed amongst the parallel signal paths of the modal resonators. For computational simplicity, since each filter $R_m(z)$ is narrowband in nature, with pole angles at $\pm\Theta_m$, we choose to approximate the feedforward and feedback paths by a delay and scalar gain. This is illustrated in Figure 4.15, where the effect of in-loop filters has been approximated for each parallel path by an integer delay and fractional delay (representing the corresponding mode's phase delay – or other measure of propagation length – from the cascade of filters in the feedback path) and a scalar gain, g_m (the gain of the lossy loop filter, $G(z)$, at the modal frequency – that is, the per-loop multiplicative loss of the mode). Recognizing the cascade of each feedforward filter and resonator as a band-limiting filter, we have thus arrived at a BWG model, with the band-limiting filters outside of the feedback path and the modes de-coupled (feeding back into themselves). Note that with the addition of a fractional delay, these band limiting filters are an additional instance of the general filter structure described in [145].

The impulse response of each BWG is an exponentially decaying sinusoid (exactly equal to the impulse response of the resonator, $R_m(z)$). Where delay is implemented by integer delay elements only, the impulse response of the BPFs (each made from a feedforward filter and resonator) is an exponentially decaying sinusoid on $t \in [0, L_m - 1]$,

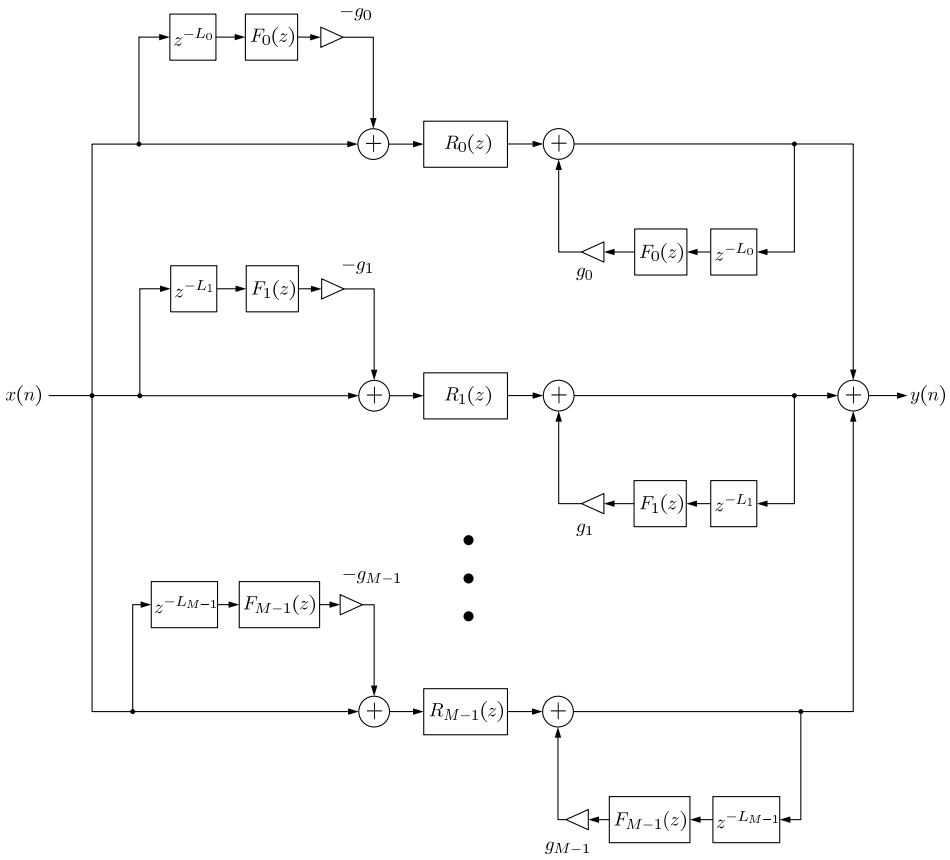


Figure 4.15: The proposed implementation of a banded waveguide model.

and zero elsewhere. When the impulse response of one of these BPFs is sent into the feedback delay loop, it will be delayed by L_m and scaled by $g_m = e^{-\alpha L_m}$ (α is the mode's decay), effectively "stitching" together the exponentially decaying sinusoid. This is not exactly the case when fractional delay filters are used, but use of the same fractional delay filters in the BPF and feedback loop ensures the correct impulse response.

If all of the resonant modes are retained in this model, the impulse response is exactly identical to that of the SDL model in Figure 4.12. Even with the approximations made, the feedforward and feedback loops in this model cancel each other out, and the resonator, derived from the original model describes the mode's impulse response.

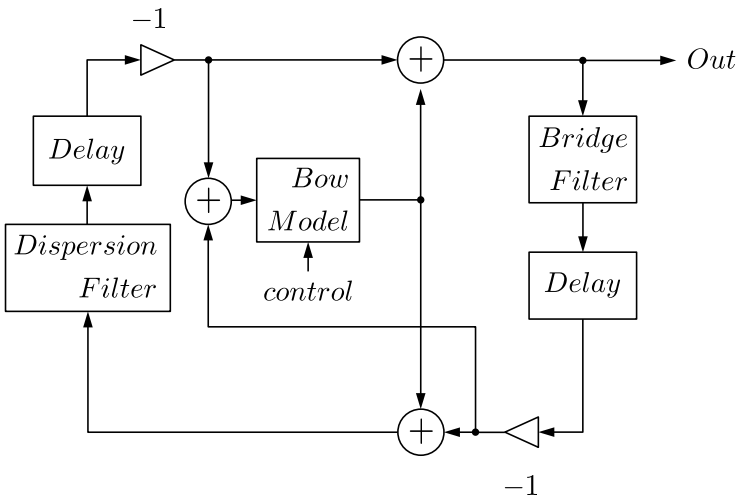


Figure 4.16: A simple bi-directional DWG model with lumping of elements and simplification of the reflection at the nut. (After models in [2, 3].)

Indeed, any choice of a filter structure and its inverse before and after the resonating filter would ensure the correct impulse response; however, use of the feedback delay loop maintains the physical interpretation of a propagating wave, permitting interactions such as bowing.

Extensions of this derivation procedure to bi-directional models, as well as possible variations in designing such BWG models are discussed in the following sections.

4.4.2 Bi-Directional Implementation

A simple bi-directional DWG model is shown in Figure 4.16 (and further described in Section 2.4.3). Note that the delay from the excitation point to the bridge has been commuted to the other side of the bridge filter for simplicity in implementation; this will result in an output that is equivalent, but advanced in time. The previously described BWG derivation procedure may be adapted here by following the same procedure of introducing an identical system and its inverse, performing a PFE, and distributing elements.

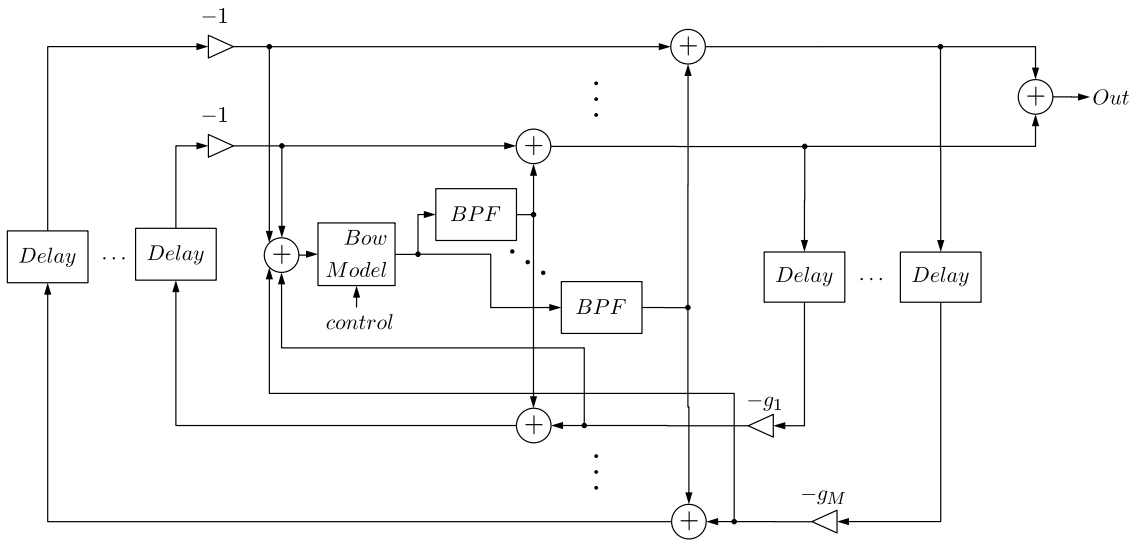


Figure 4.17: An equivalent BWG model of the bowed string. The BPF blocks may be derived by the described methods.

An example bi-directional BWG model is shown in Figure 4.17. As with previous BWG implementation, lossless propagation is modeled with delay, and losses by scalar gain. Inversion of the DWG model (for BPF design) may be carried out by using an SDL [4] decomposition, splitting the DWG model of Figure 4.16, into a feedforward excitation position filter and a string loop. Note that for stability, the nut side should actually have some loss (such that the nut-side gain is less than one for all frequencies).

One simpler approach to designing a bi-directional BWG is to use only the system's string loop and the inverse of the string loop, in place of the full SDL decomposition and inverse, in designing the BPFs. Where approximation of propagation and loss is made, this will not result in a BWG system with an impulse response identical to an equivalent DWG model. However, use of the string loop only removes the influence of interaction position (which does not affect modal frequencies or decays) in the design of

the BPF. Thus, changes in interaction position may be handled by only adjusting position in the propagation-modeling portion of models.

4.4.3 Further Variations

Using the general procedure given above to design BWG models, variations in implementation may be made. With BWG modeling intended as an efficient synthesis method, the degree of approximation in model design, as in Section 4.4.1, is subject to choice. If computational complexity was of no concern, one could create a model by the above procedure (using an entire model, instead of just a string loop), *sans* the step of approximating propagation paths with only delay and gain, that produced, within precision limits, identical results to a DWG model, even under nonlinear excitations.

Lossless propagation modeling, implemented by regular digital delays, dispersive allpass filters, and fractional delay filters, presents a number of design choices to simulate accurate wave travel. While phase delay is perhaps the most important parameter in such propagation modeling [146], group delay, transients and other various properties of a digital propagation model may affect the output. While the implementations here focus on specifying phase delay to be exact at the modal frequency of interest, more complex models could further specify group delay at the frequency of interest. Additionally, these and other properties could be optimized within a frequency band about the modal frequency. Since decaying sinusoids contain energy not just at the modal frequency, but around them, the entire frequency response of a BWG may have some effect on the synthesized output. This is further discussed in Appendix B.

In a similar fashion, loss in BWGs need not be modeled by a scalar gain set to the round-trip loss of a disturbance at the modal frequency of interest only. Filters that better capture frequency-dependent losses in a suitable frequency band may offer improved

results. Use of such filters, or more complex propagation modeling filters, however, can increase the complexity of both BWG design and implementation.

Beyond choice of how to implement the various properties of a BWG model, there is cause in variation of certain properties themselves, such as delay, dependent on the intention in model design. When designing a BWG model to be equivalent to a DWG model, if the respective phase delays of the BWG delay lines are set to that of the DWG model at the modal frequencies, then the modeled waves will not meet themselves exactly in phase. This is due to the "mistuning" of DWG models, described in detail in Appendix A, whereby the modal frequencies are influenced by the loss characteristics. Motivated by the principle of wave train closure, one may justify alternatively setting the phase delay of a BWG's delay line to be equal to an integer multiple of wavelengths of the modal frequency. In a simple KS-type BWG model, with all propagation and loss in the feedback path, as in Section 4.4.1, the impulse response of each BWG is unaffected by the implemented phase delay since the feedforward and feedback paths cancel one another. However, in the case of nonlinear or distributed interactions, differences may arise.

By the principle of closed wave trains, propagation lengths in BWG models should be chosen based on the wavelength of the modal frequency. The near-closure of wave trains in DWGs primarily arises from the approximations and digitization in design of DWG models themselves. Therefore, if one wanted, for example, to model a string with no stiffness (i.e., a string with harmonic spectra) and a fundamental frequency of 441 Hz with BWGs, the BWG parameters could be drawn directly from an analog physical model, rather than from a DWG model. However, when comparing the performance of equivalent BWG and DWG models, in the interest of emulating the DWG model, tuning of delay lengths in the BWG model to the phase delay of the DWG model may be

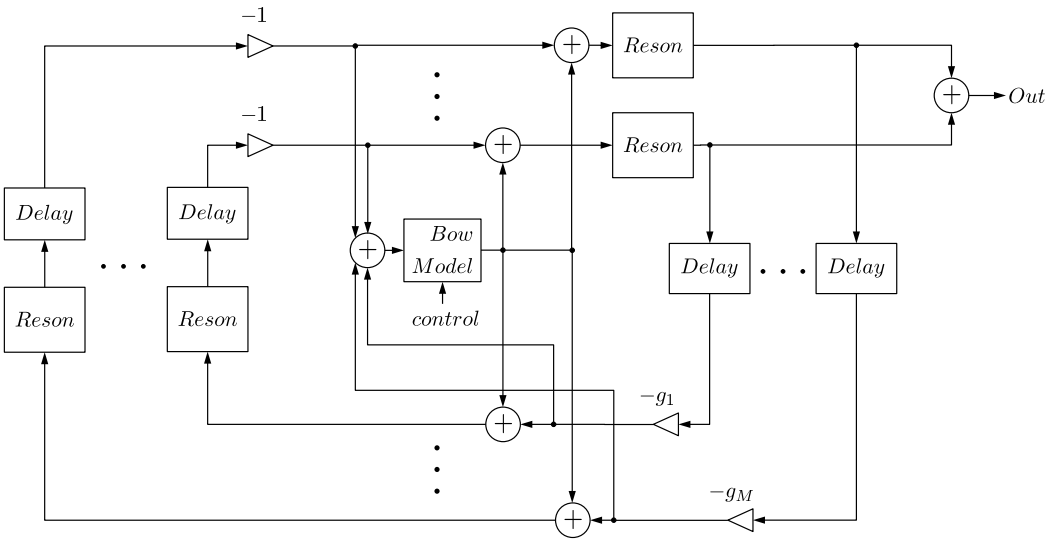


Figure 4.18: An equivalent BWG model of the bowed string, after [6].

justified. Simulations in Section 4.4.5 explore these implementation variations in delay length, comparing equivalent BWG and DWG models.

4.4.4 Comparison to Previous Implementations

To compare BWGs derived from DWG models with previous implementations of BWGs, this section presents simulation results of bowed bi-directional BWG and DWG models, using the simulation of the DWG as an ideal. Comparison of linear interactions is omitted since the BWG models proposed here may be made to have identical responses as DWG models, as is obvious from derivation (cf. Section 4.4.1).

The object model here is of a bowed string with no stiffness tuned to 441 Hz. The DWG model is implemented as in Figure 4.16 with no allpass dispersion filter, and losses modeled by a second-order linear phase filter. The BWG models are designed using the ideal harmonic modal frequencies. The DWG-inspired BWG model is designed using the string loop-only topology for BPF design. The bi-directional implementation of a reson-based BWG model is shown in Figure 4.18 [6].

Simulations of these models appear in Figures 4.19 to 4.25, showing velocity envelopes, steady-state waveforms, and spectrograms for the simulation of a string tuned to 441 Hz. The velocity envelopes are calculated by finding local minima and maxima in the time-varying waveforms. The steady-state oscillations show two pseudo-periods of the waveforms in the "steady-state," where there is little change in the output. Note that we have omitted units of velocity, as this is a synthetic example, with arbitrarily chosen values. The bow model uses the table method of [67] as implemented in the Synthesis Toolkit [124, 147].

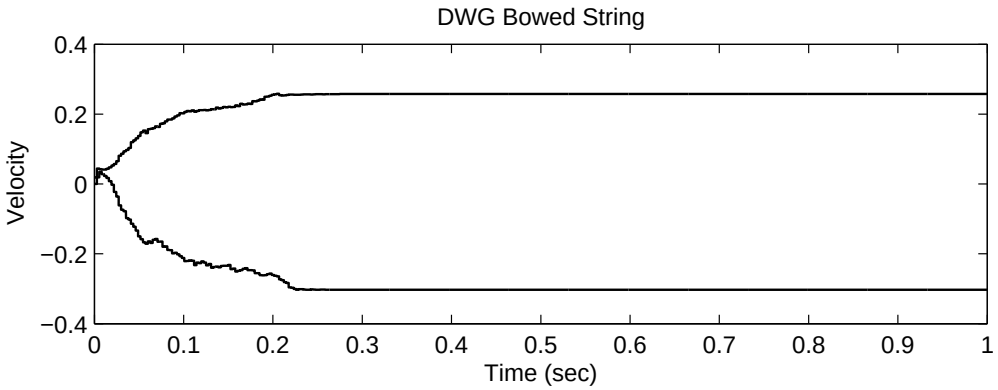


Figure 4.19: The velocity envelope of the DWG model's output.

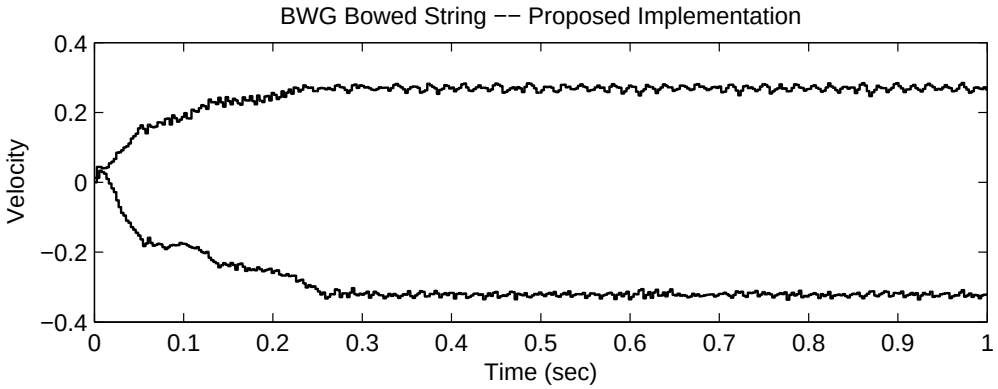
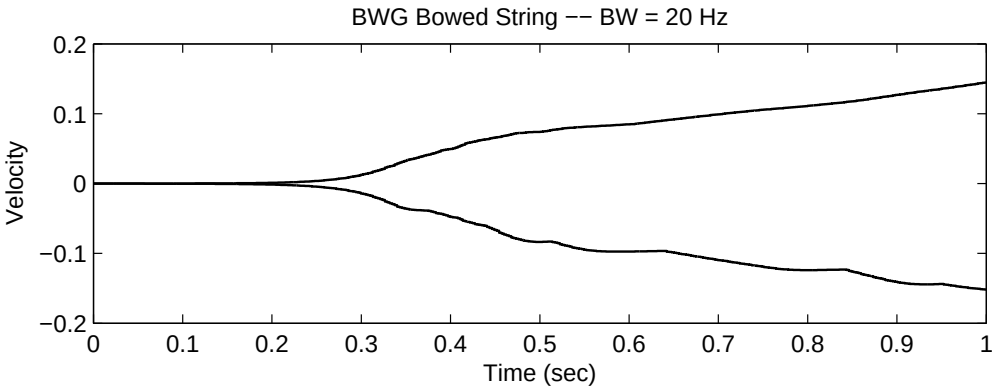
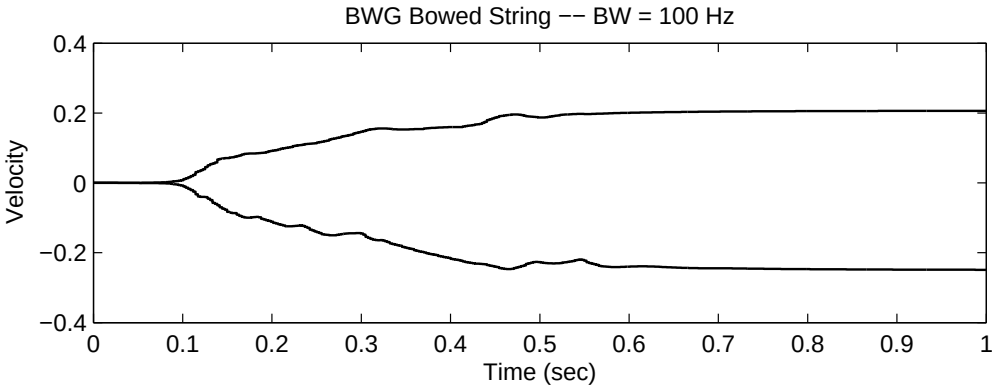


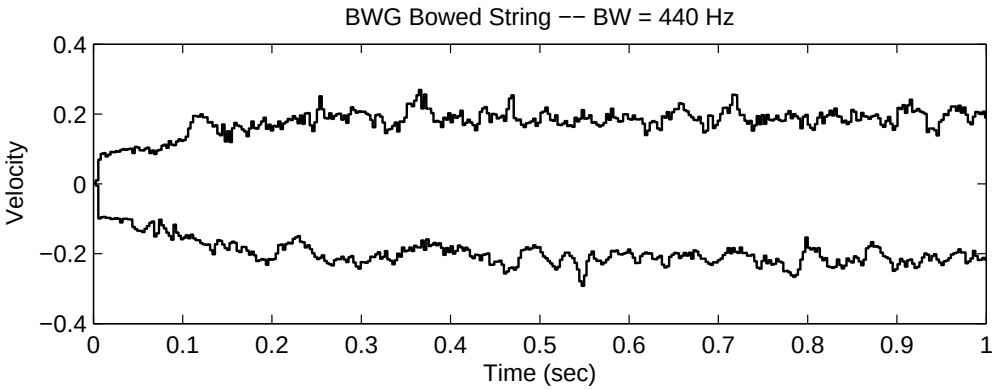
Figure 4.20: The velocity envelope of the proposed BWG model's output.



(a) 20 Hz bandwidth BPFs.



(b) 100 Hz bandwidth BPFs.



(c) 440 Hz bandwidth BPFs.

Figure 4.21: The velocity envelope of a BWG model's output, using a constant-gain resonator bandpass filter with varying bandwidths.

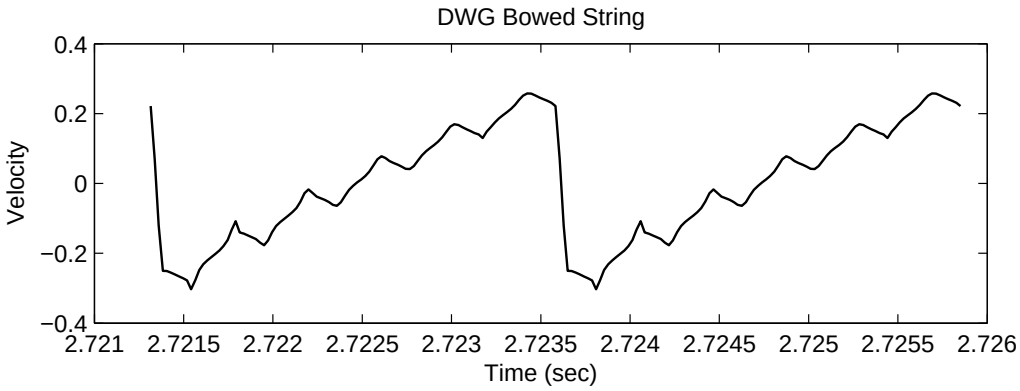


Figure 4.22: Steady-state oscillations of the DWG model's output.

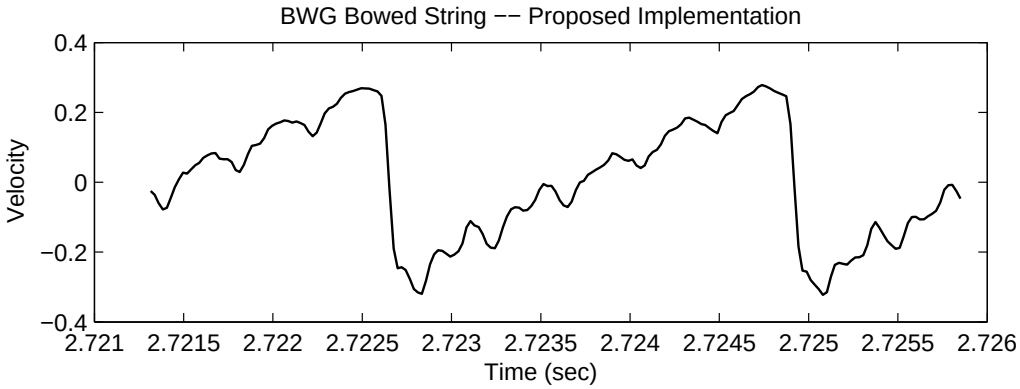
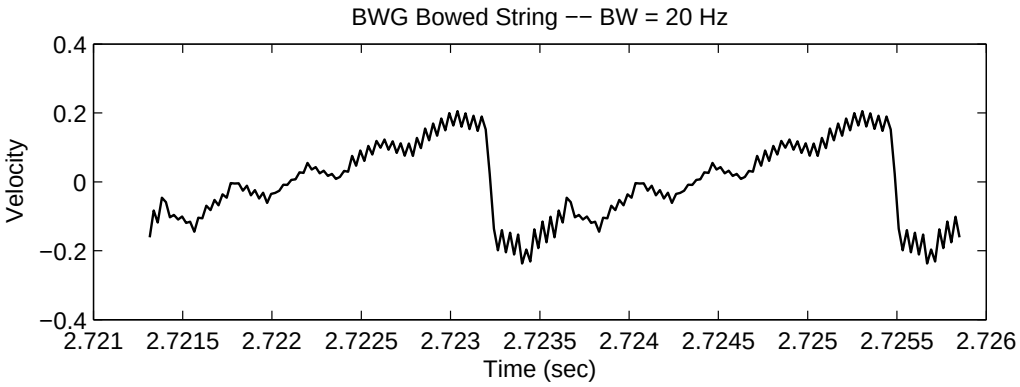
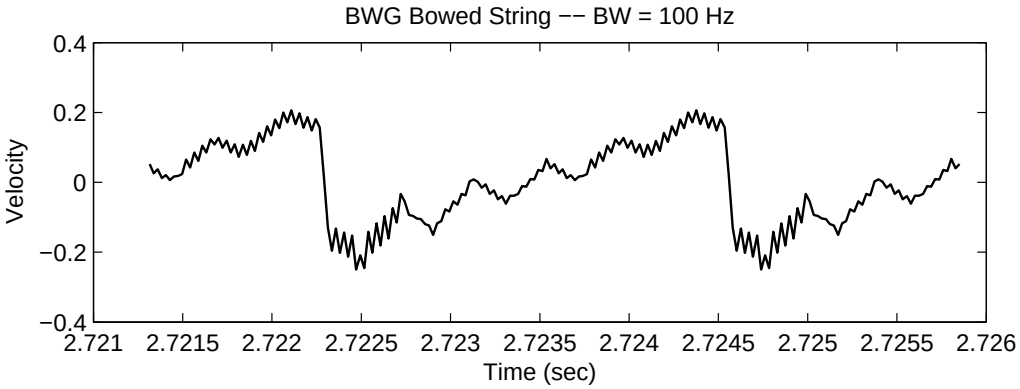


Figure 4.23: Steady-state oscillations of the proposed BWG model's output.

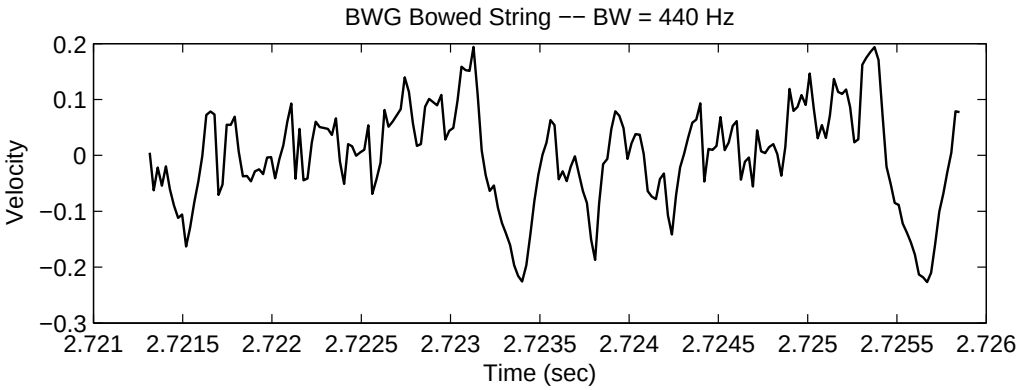
Clearly, the model presented here better simulates the attack of the DWG model, whereas many bandwidth choices for conventional BWG implementations result in slow attacks. The steady-state oscillations show noticeable differences in all BWG implementations. Listening to the resulting waveforms, there is a clear difference between the output of the DWG model and the proposed BWG model, with the proposed model exhibiting a "harsh" timbre and much energy between modes. Many simulations of the conventional BWG model yield obviously undesirable results, but some appear to sound acceptable once the steady-state is reached.



(a) 20 Hz bandwidth BPFs.



(b) 100 Hz bandwidth BPFs.



(c) 440 Hz bandwidth BPFs.

Figure 4.24: Steady-state oscillations of a BWG model's output, using a constant-gain resonator bandpass filter with varying bandwidths.

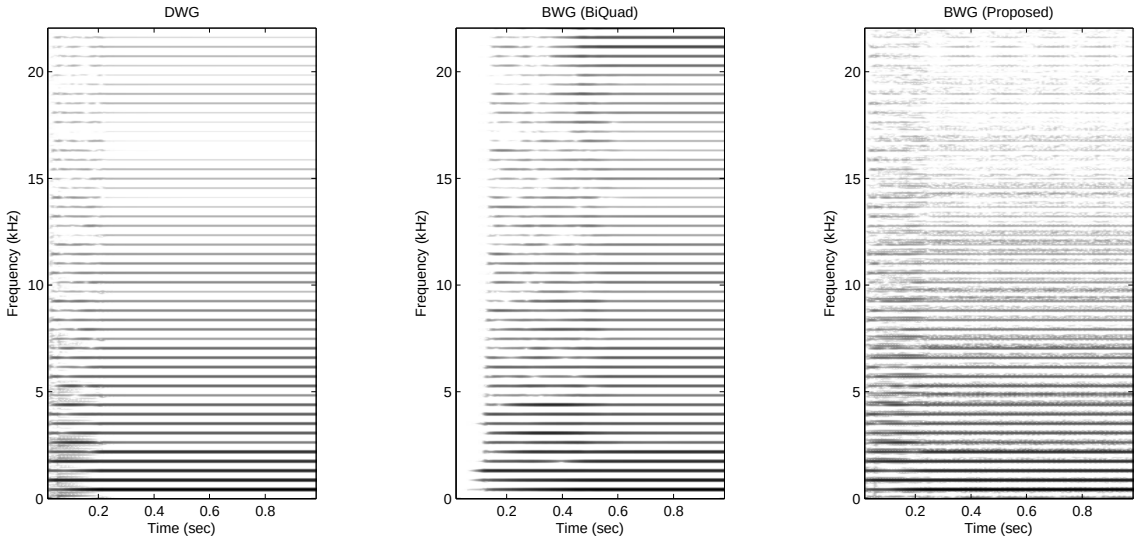


Figure 4.25: Simulation spectrograms. The biquad-based BWG model shown is that with a 100 Hz bandwidth, chosen as the best sounding biquad implementation. Black represents 0 dB; white represents -60 dB and below. Note the vibrato-like effect using the proposed topology, as well as the improved attack characteristics.

4.4.5 Simulations

To examine the behavior of variation in model and performance parameters, this section presents simulation results of variably designed DWG-inspired BWG models. Bowing of a simple DWG string model, as in Figure 4.16, is compared to comparably designed bowed BWG models. The variations in BWG model are labeled as follows:

Type A The BPFs are designed using just the string loop.

Type A1 Phase delay is set to correspond to a multiple of each modal frequency's wavelength.

Type A2 Phase delay at each modal frequency is equal to that of the DWG.

Type A3 Parameters, including modal frequency, are set to correspond to an ideal harmonic model.

Type B The BPFs are designed using the string loop and excitation position filter.

Type B1 Phase delay is set to correspond to a multiple of each modal frequency's wavelength.

Type B2 Phase delay at each modal frequency is equal to that of the DWG.

Type B3 Parameters, including modal frequency, are set to correspond to an ideal harmonic model.

All bowing simulations use a simple memoryless "bow table" method that ignores hysteresis [2]. The method implemented for bowed strings in STK [124, 147], and described in [148] is used here. This method models the junction of the bow and medium as a nonlinear, signal-dependent scattering junction [67]. The absorption coefficient, μ, of this junction may be calculated as a function of the differential velocity of the bow and the waves entering the junction , as well as the pressure of the bow, by the formula,

$$\mu(\Delta v, p) = \min\left([\,|\Delta v(5-4p)| + 0.75]^{-4}, 1\right). \tag{4.12}$$

As many of the simulations compare digital-domain models, we neglect the units of pressure and velocity, since physical properties are consistent between models. Bow velocity is dynamically changed from the start by an envelope, as in STK [124, 147].

441 Hz String with no Stiffness

Here, a simple bowed string with no stiffness is modeled. This is achieved using a DWG model as in Figure 4.16, without a dispersion filter and a near-unity gain (i.e., 0.9999) at the nut. Bowing occurs at a point 0.14 along the string, where 0 indicates the location of the bridge, and 1, the nut. As the string is tuned (at the fundamental) to 441 Hz, with a sampling rate of 44.1 kHz, the delays in the DWG model are all pure integer delays. The bridge filter is implemented with a second-order linear phase filter. The choice of tuning, bow position, and stiffness allows the BWG models (excepting Type A1 and B1) to also be implemented with integer delays only, so as to prevent any possible effects due to choice of fractional delay filters.

The simulation results, including amplitude envelopes, spectrograms, and plots of steady-state pseudo-periods appear in Appendix C.1. It may be seen that BWG Types A1 and A3 give very similar results. They more often display a ramp-like waveform as in the DWG model, as compared to the other BWG types. The amplitude envelopes also more closely follow that of the DWG model. The Type B models produce simulations that seem to significantly differ from the DWG model, with the amplitude envelope often growing to values well beyond that of the DWG model's.

The results from this example seem to suggest that use of the string loop only in designing BWGs may be best. Additionally, the results indicate that phase closure, as in Types A1 and A3, may be an especially important feature in BWG models. This, of course, aligns with the perspective of BWGs modeling closed wave trains [6, 7, 70]. As with the simulation of Section 4.4.4, the BWG outputs here exhibit a "harsher" timbre as compared to the DWG, with significant spectral energy between modes.

441 Hz String with Stiffness

Here, a bowed string model is implemented as in the previous section, but with a second-order allpass dispersion filter and the fundamental tuned via setting phase delay at 441 Hz. The simulation results, including amplitude envelopes, spectrograms, and plots of steady-state pseudo-periods appear in Appendix C.2. In this example, the Type B BWGs designed using the full SDL decomposition, as in the non-stiff example produce steady-state oscillations that do not appear to resemble (in any of the cases), the DWG model. However, the amplitude envelopes are better in this case. Still, the results from the Type A BWGs more closely resemble that of the DWG model. As with previous results, the BWG simulations create "harsher" timbres. Interestingly, the disparity between the Type A1 and Type A2 BWG models is greatly reduced here.

4.5 Concluding Remarks on Object Re-Sonification

While the newer DWG-derived BWG models may exhibit better attack characteristics in the given simulations of nonlinear excitation, previous implementations seem to currently offer the possibility of better sounding steady-state outputs. This, however, requires careful choice of bandwidth for the biquad BPFs. As discussed in Section 4.4.3, the BWG model described here could be adapted to give a faithful reproduction of a DWG model by skipping approximating steps, but at great computational expense, defeating the purpose of BWG models. This warrants further study of the effect of approximating steps in the design of such BWG models.

Choice of BWG model for applications may be best to consider on a case-by-case basis. For many musical applications, timbre of the steady-state oscillations may be an important feature, and nonphysical variations may even be aesthetically pleasing. Where more physically-based accuracy is important, use of the DWG-derived models seems appropriate, and is certainly so for linear interactions at a point.

Chapter 5

RE-SONIFICATION OF EVENTS

This chapter is concerned with the re-sonification of events, contextually defined as the imparting of energy into a sounding object. Taking a source/filter view of events with objects, we are thus concerned with estimating the source, i.e., the input or excitation to sounding object models. With a separation between an object and the event that excites it, separate transformation or exchange of either component is possible. Methods of modeling, estimating, and transforming excitations are briefly reviewed, followed by a novel approach to estimate percussive excitations. This percussive excitation estimation method relies on linear estimation of system inputs, constrained to a signal-subspace. The original work herein was first presented and described in [10] and [11].

5.1 Modeling and Transforming Events

Modeling of exciting events is dependent on the object model used. With digital physical models, events may be modeled by digitization of the exciting phenomenon. In practice, a measure of approximation may be made. With DWG models, for example, plucking and hammering may be simulated by loading delay units appropriately to model each event's effect on a string, or by more complex interaction models [2]. Interactions such as bowing may be implemented by simulating bow-string dynamics [63,64,149], including any number of simplifications [2,3]. Even arbitrarily chosen or misappropriated interaction models may be applied to object models for aesthetic purposes [150].

Viewing physical, pseudo-physical, or other models as filters, one may interpret inputs as exciting events. Such input signals may be physically motivated, empirically estimated, arbitrarily chosen, or a mixture of these . Consider, for example, commuted piano synthesis, where the input to a string model includes both the soundboard response and a signal-based approximation of hammering [2, 151, 152]. A recent example of exci-

tation modeling may be found in [153], where guitar excitations are estimated from data and statistically parameterized for expressive and transformative resynthesis.

More perceptually-based spectral (and related) models easily permit temporal and spectral transformations [39, 40, 79–82, 91] and methods including transient modeling [39, 40, 91] are especially suited to capturing impulsive or wide-band exciting events. Alterations, though, may not illicit perception of event transformation with an invariant object in the absence of explicit consideration of physical or perceptual phenomena. Thus, physically-based and hybrid models are especially suited for transformation and exchange of events, where there is meaningful consideration of the influence of both events and objects in forming sound [154].

5.2 Estimating Excitations

For purposes of re-sonification, exciting events may be estimated from observed sounds, for identity resynthesis, event transformation, or use with differing object models. The task of excitation estimation may be viewed as an application of input estimation, where the output (i.e., a sound recording) has been observed, usually subject to noise. Thus, one seeks whatever *information* is necessary to reconstruct the observed sound using some object model [91]. Herein, methods for excitation estimation assuming an exciter/resonator model are assumed, where the resonating object model is known.

Inverse filtering is one such method for recovering inputs [2, 14, 114]. In a digital filter model, this constitutes exchanging zeros for poles, and vice versa. (Minimum-phase zeros are required for stability.) Inverse filtering of the output signal will yield a residual, which may be used to accurately produce the observed output [2]. However, if a recording is noisy, estimation of the residual excitation signal will be highly corrupted by noise, yielding undesirable results. Further extensions to inverse filtering for excita-

tion estimation are used in [115], where the output is split into sinusoidal and residual portions.

Least-squares (LS) estimation provides an alternative to inverse filtering, using an approach based in linear estimation theory to account for the presence of noise. Sound excitation estimation by LS has been used in [138], where multiple recordings of the same sound type (a piano note in this case) are aligned and used to find a single estimate of the excitation. LS estimation finds an estimate of the input signal, that, when it excites the model of interest, produces an output that is "closest" to the observed output (minimizing the ℓ^2-norm of the error). These and related methods are further described in [155].

5.3 Estimating Percussive Sound Excitations

Percussive sounds generally include those produced by a short-lived impact. For example, a struck marimba bar or hammered piano string will produce a percussive sound. Even many non-percussive sounds, such as that of a plucked guitar, might be considered percussive-like, due to their limited duration of excitation. Considering the time-limited nature of percussive sound excitations, one may use this knowledge in resonifying percussive sounds. Note, also, that in estimating modal parameters from sound recordings, percussive excitations are desirable due to their wide-band frequency content and short-lived nature, resulting in large portions of recorded percussive sounds being dominated by the resonant modes of an object.

The short duration of percussive excitations is often considered for resonification. In [2, 36, 138], excitation estimates are truncated after decaying to sufficiently small amplitudes. The estimation procedures used, however, do not explicitly consider the limited duration of percussive sound excitations. We have therefore proposed a method of estimating percussive sound excitations which constrains the estimates

to be limited in time [10]. This is presented as a special case of estimating inputs constrained to a signal-subspace [11].

5.3.1 Signal Subspace-Constrained Input Estimation

We consider single-input single-output (SISO) discrete-time modal state space systems of the form

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{G}_{t+1} u_{t+1} \tag{5.1}$$

$$y_t = \mathbf{H}_t \mathbf{x}_t + v_t, \tag{5.2}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the unobserved state vector at time t, u_t is the unknown excitation signal, and y_t is the observed sound or output including the measurement noise, v_t. The system is a state space equivalent of parallel second-order resonators [2], with $\mathbf{x}_t$ representing the collection of the internal states of each resonator. We assume zero-mean white Gaussian noise, although the described method will apply to any zero-mean white noise process without alteration. The $n \times n$ state matrix, $\mathbf{F}_t$; the $n \times 1$ input vector, $\mathbf{G}_t$; and the $1 \times n$ output vector, $\mathbf{H}_t$, are assumed to be known or estimable. We then consider the problem of estimating the excitation signal, u_t, for all observed times, $t \in [1, T]$, based on the noisy sound, y_t.

Where u_t is constrained to some known signal subspace, it may be represented by the basis expansion,

$$u_t = \sum_{i=1}^{L} \gamma_i(t) u_{b,i} \,, \tag{5.3}$$

where the L basis signals, $\{\gamma_1(t), \gamma_2(t), \ldots, \gamma_L(t)\} \; \forall \, t \in [1, T]$, span the input signal subspace of interest, and each *time-invariant* coefficient, $u_{b,i}$, is a scalar constant. This type of constraint for input estimation was proposed in [156] and expanded upon in [11]. Using (5.3), with knowledge of the basis signals, the excitation signal may be reconstructed

from the coefficients, and an estimate of the excitation signal may be obtained by using estimates of the coefficients.

Using (5.3), the state update equation, (5.1), may be changed to

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{G}_{b,t+1} \mathbf{u}_b, \tag{5.4}$$

where

$$\mathbf{G}_{b,t} = \mathbf{G}_t \mathbf{\Gamma}_t, \tag{5.5}$$

$$\mathbf{\Gamma}_t = \begin{bmatrix} \gamma_1(t) & \gamma_2(t) & \dots & \gamma_L(t) \end{bmatrix}, \tag{5.6}$$

$$\mathbf{u}_b = \begin{bmatrix} u_{b,1} & u_{b,2} & \dots & u_{b,L} \end{bmatrix}^T. \tag{5.7}$$

The time-variance of the input is then represented in the newly formulated input matrix, $\mathbf{G}_{b,t}$, to which the time-varying portion of the excitations signal's representation, $\mathbf{\Gamma}_t$, has been moved.

To express the observed output directly as a function of the coefficient vector, $\mathbf{u}_b$, we consider the alternative system formulation (assuming $\mathbf{x}_0 = \mathbf{0}$),

$$\mathbf{C}_{t+1} = \mathbf{F}_t \mathbf{C}_t + \mathbf{G}_{b,t+1} \tag{5.8}$$

$$y_t = \mathbf{H}_t \mathbf{C}_t \mathbf{u}_b + v_t \tag{5.9}$$

where

$$\mathbf{C}_1 = \mathbf{G}_{b,1}. \tag{5.10}$$

The collection of all observed outputs at time t may then be represented as

$$\mathbf{y}_{1:t} = \mathbf{A}_{1:t} \mathbf{u}_b + \mathbf{v}_{1:t}, \tag{5.11}$$

where

$$\mathbf{y}_{1:t} = \begin{bmatrix} y_1^T & y_2^T & \dots & y_t^T \end{bmatrix}^T, \tag{5.12}$$

$$\mathbf{v}_{1:t} = \begin{bmatrix} v_1^T & v_2^T & \dots & v_t^T \end{bmatrix}^T, \tag{5.13}$$

$$\mathbf{A}_{1:t} = \begin{bmatrix} \mathbf{H}_1\mathbf{C}_1 \\ \mathbf{H}_2\mathbf{C}_2 \\ \dots \\ \mathbf{H}_t\mathbf{C}_t \end{bmatrix}. \tag{5.14}$$

A LS estimate (the best linear unbiased estimate) of $\mathbf{u}_b$, represented at time t by $\hat{\mathbf{u}}_{b,t}$, is then the solution to

$$\hat{\mathbf{u}}_{b,t} = \arg\min_{\mathbf{u}_{b,t}} ||\mathbf{y}_{1:t} - \mathbf{A}_{1:t}\mathbf{u}_{b,t}||^2. \tag{5.15}$$

Calculation of this estimate may be done off-line via solution of the associated normal equation, or estimates may be obtained at each time step by using a recursive least-squares (RLS) algorithm [11]. The estimate of the entire excitation signal may be reconstructed at any time by substituting the estimates for the $u_{b,i}$ in (5.3). Note that the computational complexity of estimation is dependent on the chosen basis expansion and method of finding a LS solution.

Recursive Input Estimation

The associated RLS equations to estimate $\mathbf{u}_b$ are

$$\hat{\mathbf{u}}_{b,t+1} = \hat{\mathbf{u}}_{b,t} + \mathbf{K}_{t+1}\left(\mathbf{y}_{t+1} - \mathbf{A}_{t+1}\hat{\mathbf{u}}_{b,t}\right) \tag{5.16}$$

$$\mathbf{K}_{t+1} = \mathbf{P}_t\mathbf{A}_{t+1}^T(\mathbf{I} + \mathbf{A}_{t+1}\mathbf{P}_t\mathbf{A}_{t+1}^T)^{-1} \tag{5.17}$$

$$\mathbf{P}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{A}_{t+1})\mathbf{P}_t \tag{5.18}$$

where

$$\mathbf{A}_t = \mathbf{H}_t \mathbf{C}_t \tag{5.19}$$

$$\mathbf{P}_t = (\mathbf{A}_{1:t}^T \mathbf{A}_{1:t})^{-1}. \tag{5.20}$$

Note that this algorithm has the same complexity at each time step. Typical recursive algorithms for obtaining smoothed estimates (i.e., optimal estimates based on *all* observations) require a backwards smoothing step [157] or a forward filtering process that grows in complexity [158]. Where the elements of $\mathbf{u}_b$ may be directly interpreted as sample values of the input (such as in the case $\gamma_i(t) = \delta(t-i)$), the above RLS algorithm amounts to finding optimal smoothed estimates of the entire input signal with a fixed complexity forward filtering process, without need for a backwards smoothing process. The standard RLS algorithm, however, may not necessarily be applied at all times, as discussed below.

Rank-Deficient Recursive Estimation

Where $\mathbf{A}_{1:t}$, analogous to an observability matrix, does not have full column rank, the RLS algorithm is unable to calculate $\mathbf{P}_t$ in (5.20), due to the singularity of $\mathbf{A}_{1:t}^T \mathbf{A}_{1:t}$. This will always occur when estimating $\mathbf{u}_b$ at some time, τ, where, for any i, $\gamma_i(t) = 0 \, \forall \, t \leq \tau$. That is, solving for values of the input that have not occurred or been observed is ill-posed (without any further assumptions about the structure of $\mathbf{u}_b$). Consider for example, a single-input system with $\gamma_i(t) = \delta(t-i)$, such that each $u_{b,i}$ represents the input's value at time i. Estimating any $u_{b,i}$ that corresponds to a future unobserved value amounts to prediction, and with no further assumptions about the input, the matrix inversion of (5.20) may not be carried out.

To address such situations in recursive estimation, one may find the minimum-norm LS estimate of $\mathbf{u}_b$. (Use of the minimum-norm estimate motivates normalizing the energy of the basis signal vectors.) Recursive estimates of $\mathbf{u}_b$ may be found by using the

method of [159] to recursively compute the Moore-Penrose pseudoinverse of $\mathbf{A}_{1:t}^T\mathbf{A}_{1:t}$. The method of [159] recursively calculates the singular value decomposition (SVD) of $\mathbf{A}_{1:t}^T\mathbf{A}_{1:t}$ to find its pseudoinverse and obtain LS estimates of the unknown variable. With the singular values of $\mathbf{A}_{1:t}^T\mathbf{A}_{1:t}$ calculated at each step, one may easily determine when the inversion of (5.20) is well-conditioned and the more efficient regular RLS algorithm may be applied.

Mismatched Constraint

In practice, one may not have full knowledge of the signal subspace, $\mathscr{S}$, to which the input is constrained. In such cases, one may only have access to an estimate, $\hat{\mathscr{S}}$, of the signal subspace $\mathscr{S}$. In this section, we consider the effect of applying the proposed estimation procedure when $\hat{\mathscr{S}} \neq \mathscr{S}$. In these cases, in lieu of $\mathbf{\Gamma}_t$, the estimation procedure uses a mismatched BE, represented by the time-varying matrix, $\hat{\mathbf{\Gamma}}_t$, of column-wise ordered basis vectors that span $\hat{\mathscr{S}}$.

We first consider cases where the signal subspace $\mathscr{S}$ is "overestimated," i.e.,

$$\mathscr{S} \subset \hat{\mathscr{S}}. \tag{5.21}$$

In such cases, we choose to represent $\hat{\mathbf{\Gamma}}_t$ in block matrix form by

$$\hat{\mathbf{\Gamma}}_t = \begin{bmatrix} \mathbf{\Gamma}_t & \mathbf{\Psi}_t \end{bmatrix}, \tag{5.22}$$

where $\mathbf{\Psi}_t$ is a matrix of time-varying vectors not in the signal subspace of the input. That is, the input at time t is a linear combination of the columns of $\mathbf{\Gamma}_t$, but the estimation procedure assumes the input is in the signal subspace spanned by the basis signal vectors of $\mathbf{\Gamma}_t$ in addition to the extraneous basis signal vectors of $\mathbf{\Psi}_t$.

Using $\hat{\mathbf{\Gamma}}_t$ as defined in (5.22), it may be shown that the expected value of the final estimate (i.e., the estimate at time T) obtained by (5.15) is

$$E(\hat{\mathbf{u}}_{b,T}) = \begin{bmatrix} \mathbf{u}_b \\ \mathbf{0} \end{bmatrix}. \tag{5.23}$$

Thus, the estimation procedure is unbiased in such cases of "overestimation." The coefficients associated with the extraneous basis signal vectors of $\mathbf{\Psi}_t$ have an expected value of $\mathbf{0}$; the coefficients associated with $\mathbf{\Gamma}_t$ have an expected value of $\mathbf{u}_b$, the coefficients of the input signal's correct BE.

We consider the signal subspace $\mathscr{S}$ to be "underestimated" when

$$\hat{\mathscr{S}} \subset \mathscr{S}. \tag{5.24}$$

Choosing to represent $\hat{\mathbf{\Gamma}}_t$ by

$$\mathbf{\Gamma}_t = \begin{bmatrix} \hat{\mathbf{\Gamma}}_t & \mathbf{B}_t \end{bmatrix}, \tag{5.25}$$

one may interpret such cases as omitting some of the basis signal vectors of $\mathbf{\Gamma}_t$ in the estimation procedure. The input at time t may then be expressed as

$$\mathbf{u}_t = \hat{\mathbf{\Gamma}}_t \mathbf{u}_b^{\hat{\Gamma}} + \mathbf{B}_t \mathbf{u}_b^B, \tag{5.26}$$

splitting the input into contributions from the selected basis signal vectors and the omitted basis signal vectors. In this case, $\hat{\mathbf{u}}_b$ is then an estimate of $\mathbf{u}_b^{\hat{\Gamma}}$.

Given (5.25), the outputs may be expressed by

$$\mathbf{y}_{1:T} = \mathbf{A}_{1:T}^{\hat{\Gamma}} \mathbf{u}_b^{\hat{\Gamma}} + \mathbf{A}_{1:T}^B \mathbf{u}_b^B + \mathbf{v}_{1:T}, \tag{5.27}$$

where $\mathbf{A}_{1:T}^{\hat{\Gamma}}$ and $\mathbf{A}_{1:T}^B$ represent the matrix of (5.14), calculated using $\hat{\mathbf{\Gamma}}_t$ and $\mathbf{B}_t$, respectively The expected value of the estimated basis coefficients is then

$$E(\hat{\mathbf{u}}_{b,T}) = \mathbf{u}_b^{\hat{\Gamma}} + \mathbf{A}_{1:T}^{\hat{\Gamma}\dagger} \mathbf{A}_{1:T}^B \mathbf{u}_b^B, \tag{5.28}$$

where (†) indicates the Moore-Penrose pseudoinverse. Thus, $\hat{\mathbf{u}}_{b,T}$ may be a biased estimate of $\mathbf{u}_b^{\hat{\Gamma}}$, depending on the value of $\mathbf{A}_{1:T}^{\hat{\Gamma}\dagger}\mathbf{A}_{1:T}^{B}\mathbf{u}_b^{B}$.

Application to Estimating Percussive Sound Excitations

Applied to the estimation of percussive sound excitations, the signal subspace constraint may equivalently be seen as a constraint on the duration of the excitation. Aligning the start of the excitation at time $t = 1$, we may then use, among other possible choices, the basis signals,

$$\gamma_i(t) = \delta(t - i), \tag{5.29}$$

such that each $u_{b,i}$ is then the sample value of the excitation at time $t = i$. The maximum duration, L, of the excitation is assumed to be known *a priori* or estimable from the recording. Even when the duration is not known, overestimating can still give improved results, as shown below.

5.3.2 Evaluation of Synthetic Data

To assess the performance of the described constrained estimation method, consider a synthetic example where the system model and excitation duration are known. The system model used is a time-invariant lumped modal model of an A4 (440 Hz) piano string. An excitation signal of 20 ms of exponentially decayed white noise was applied to the model, from which a clean sound was synthesized. Simulated noisy recordings were then generated using additive white Gaussian noise (AWGN), and the excitation signal was estimated. The constrained estimation method was used, along with unconstrained LS estimation [138], and inverse filtering [2, 14]. Since we are considering a case where the duration of the excitation is known *a priori* or estimable, all estimates are truncated to the known duration. Typical excitation estimation methods for resynthesis employ truncation after estimation, based on energy levels [2, 14]. This analysis additionally considers the mismatched case where the duration of the excitation is overestimated to

40 ms, as durations used in practice are likely to be estimates or heuristically determined values.

Comparing the results obtained with the constrained method and unconstrained LS, the excitation estimates themselves appear quite similar in the time domain, with our method often having a marginally smaller mean square error (MSE). However, the constrained method gives improved estimation of the excitation signal in bands about the model's resonant frequencies; the effect of this becomes discernible in the resynthesized signal. The inverse filter residual estimate was found to be less accurate than the other estimators; this may be in part due to an inexact inverse filter being used for guaranteed stability [2].

As the considered example presents the greatest discernible difference between methods at the resonant modal frequencies, and since transient portions of the signal are difficult to accurately reproduce without a high signal-to-noise ratio (SNR), we examine the spectra of the resynthesized signals about the modes. Inaccurate estimation of the excitation energy at the resonant frequencies results in a noticeable timbral difference between the resynthesized and original signals. To quantify this effect, we use a spectral distance measure we call the Modal Log-Spectral Distance (MLSD), defined as

$$MLSD = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left[10\log_{10}\left(\frac{\hat{P}(f_m)}{P(f_m)}\right)\right]^2}, \tag{5.30}$$

where P and $\hat{P}$ are the power spectral density of the original and resynthesized signals, respectively, and each f_m is one of the M modal frequencies. This is similar to the RMS Log-Spectral Measure in [160], but evaluated only at the modal frequencies. (Note that the excitation's short duration ensures that it has little effect on the decay of the modes.) The resynthesized signals are scaled such that the MLSD is minimized, as we are interested in comparing the modal amplitudes relative to one another, and not the overall

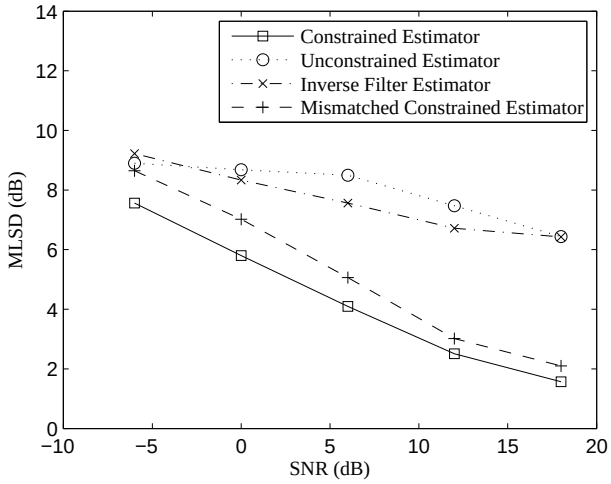
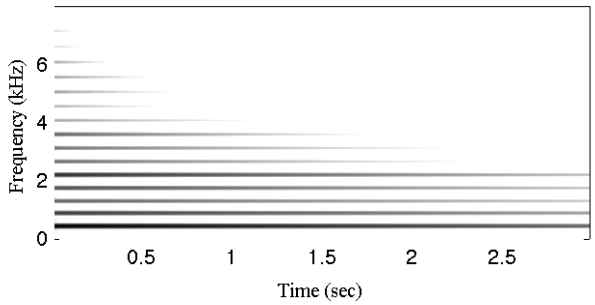


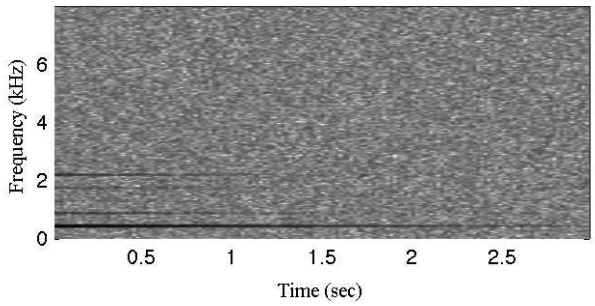
Figure 5.1: Average MLSD of resynthesized signals, as compared to the original noiseless signal, using different excitation estimates. Use of the proposed constrained estimator, even with a mismatched duration, gives improved results over unconstrained LS estimation and inverse filtering. (100 simulations per point.)

volume. Figure 5.1 shows the average MLSD for the resynthesized sound using the excitations estimated with the constrained method, unconstrained LS, and inverse filtering, with the analyzed signal at various SNRs. Additionally, Figure 5.1 shows the result of resynthesis using a mismatched constraint, overestimating the excitation duration to be doubled (40 ms). These results show an improvement using the constrained estimation method, including the case where the constraint is mismatched. Not shown in Figure 5.1, our method also gave an improved MLSD over the other estimation methods when their estimates were truncated to 40 ms.

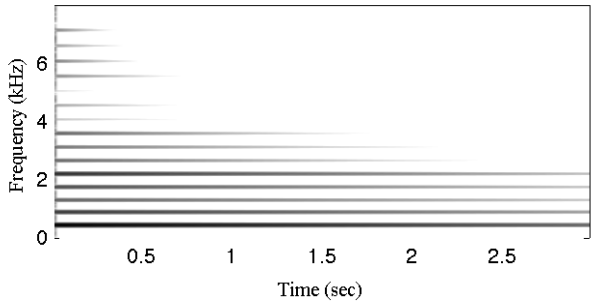
To qualitatively confirm this estimation method's use for resynthesis, informal listening tests were performed to evaluate a set of results from the previously described synthetic piano example. Excitations were estimated from a noisy sound example with an SNR of -6 dB using the previously described methods; these excitations were then used to create resynthesized sounds. Two participants were played pairs of resynthesized



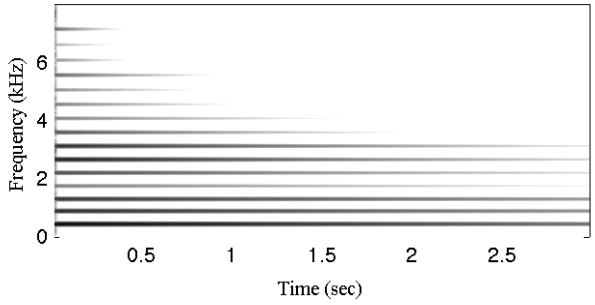
(a)Original (clean) synthesized signal.



(b)Original signal in noise (-6 dB SNR).



(c)Resynthesized signal using our excitation estimation method.



(d)Resynthesized signal using the unconstrained LS excitation estimate.

Figure 5.2: Normalized signal spectrograms. Black indicates 0 dB; white indicates -60 dB and below.

sounds and asked which was most similar to the original signal without noise. The sound created using the constrained estimate (with the correctly matched duration) as the excitation was chosen over the sounds using the other estimated excitation signals in all but one trial; in two repetitions of the same trial, however, the participant chose the sound made with our proposed estimate. Participants noted difficulty in choosing between sounds using the matched and mismatched constrained estimator, further indicating the utility of our method, even with some mismatch in duration. Spectrograms of the original signal, the noisy signal, and the resynthesized signal, using the constrained method and unconstrained LS, are shown in Figure 5.2, for visual comparison of the time-frequency characteristics of the signals. The difference in the levels of the 6th and 7th harmonics of the two resynthesized signals, as compared to the original signal, is perhaps most noticeable. These sounds may be heard at [161].

5.3.3 Evaluation of Recorded Sounds

When applying this method to recorded sounds, the modal system models and duration of the excitation may not be known. In these cases, one may estimate them. Modal systems are typically estimated from data by fitting decaying sinusoids to a sound recording [2, 8, 21, 120]. Here, an exponential curve is fit to the progression of each of the mode amplitudes across successive frames.

This method has been applied to recorded sounds of percussively struck objects, recorded at 44.1 kHz, including a tuning fork and large serving fork. These sounds may be found at [161]. The excitation estimates were found assuming a 20 ms duration (chosen based on manual inspection of the sounds) and by using a modal model estimated as in [8]. Informal evaluation of these resynthesized signals in comparison to the original recordings has supported the validity of our estimation method. In the case of recordings with a very high SNR, it has been observed that there is very little discernible differ-

ence between resynthesis with the constrained estimation method and an unconstrained estimator.

5.4 Discussion

The constrained estimation method described here has been shown to be sufficient for recovery of percussive events, performing particularly well compared to similar methods when sound recordings are very noisy. This implies suitable application of this method to re-sonification of percussive sounds for de-noising purposes. In using this method, accurately estimating the duration of excitation signals is important to ensure matching the constraint to actual excitations. Using a constraint that is underestimated (too short, compared to the true duration) will bias the estimate of the excitation [11], therefore, overestimation of the duration of excitations may be preferable to underestimation. In the absence of previously derived empirical or heuristic estimates of the duration of percussive sound excitations, the duration may be estimated by transient detection [40] or examination of the unconstrained estimator [2].

In applications where the properties of the object are unknown, re-sonification could likely be improved by joint system and input estimation [162, 163], with proper incorporation of constraints. A motivation for examining percussive sounds, though, is the ease of using spectral analysis methods to examine the late response of sounding objects to estimate object model parameters (i.e., modal frequencies and decay rates) [2]. Other improvements might be had through parameterization or classification of many estimated excitations, as in [153] and [115].

Chapter 6

CONCLUSION

Approaches to the re-sonification of objects, events, and environments have been presented, including original contributions. In consideration of re-sonification at the environmental level, an approach to soundscape design and synthesis, using acoustic, semantic, social, and geographic information was described. For the modeling of objects, the use of BWGs was considered, and it was shown how to improve their implementation to more appropriately apply methods applicable to modal and DWG models. To work toward event re-sonification, specifically interactions with sound models, an original method of estimating percussive sound excitations that results in improved resynthesis, especially in the presence of noise, was presented.

As with any subject, there are still lingering questions and areas ripe for future research. With regard to constructing BWG models that more closely mimic DWG models, as mentioned in Section 4.4.3, one could make no simplifying approximations in modeling propagation to provide identical results. This would be computationally costly, but may be useful in separating the various modes for analytical or (likely offline) musical purposes. Still, a less simplified approximation of propagation paths might yield better results in some situations. This has not been extensively explored, given the number of approaches and parameters to consider, with an eye on generality. Even the proposed implementation of BWG models could be explored for further object and nonlinear interaction simulations. Application of the BWG model described here for re-sonification involving linear exciting events allows identity resynthesis, but the fidelity of nonlinear interactions may vary.

For estimating object models and events, more methods could be explored, including joint estimation methods. The described constrained estimation method relies

on a modal object model that is separately estimated or known *a priori*. Joint estimation may offer improvement in suggested de-noising applications where the object model is unknown. Additionally, application of the general estimation procedure, where the input is limited to a signal-subspace, to contexts beyond percussive excitations could be examined.

The contributions described of this work apply to but a few of the existing methods and approaches to re-sonification. One needs to only look at the plethora of existing literature on the subject, a sampling of which is referenced here, to see evidence of long-held interest and work in re-sonification. A completely general panacea for re-sonification at all scales that considers physics, perception, ecology, and other sonic perspectives may not be feasible. However, the constant increasing of computational resources and availability of information better permits improvement and unification of re-sonification methods. Various contributions such as those described here, serve to further specific applications, while moving toward generality.

REFERENCES

[1] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith III, "The simulation of piano string vibration: from physical models to finite difference schemes and digital waveguides," *Journal of the Acoustical Society of America*, vol. 114, no. 2, pp. 1095–1107, August 2003.

[2] J. O. Smith III, *Physical Audio Signal Processing*. W3K Publishing, 2010.

[3] S. Serafin, "The sound of friction: Real-time models, playability and musical applications," PhD dissertation, Stanford University, 2004.

[4] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998.

[5] G. Essl and P. R. Cook, "Banded waveguides: Towards physical modeling of bowed bar percussion instruments," in *Proc. of the ICMC*, 1999, pp. 321–324.

[6] G. Essl, S. Serafin, P. R. Cook, and J. O. Smith III, "Theory of banded waveguides," *Computer Music Journal*, vol. 28, no. 1, pp. 37–50, 2004.

[7] G. Essl, "Physical wave propagation modeling for real-time synthesis of natural sounds," PhD dissertation, Princeton University, 2002.

[8] A. Fink, H. Thornburg, and A. Spanias, "An individually tunable perfect reconstruction filterbank for banded waveguide synthesis," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, March 2010, pp. 37–40.

[9] A. M. Fink, A. S. Spanias, and P. R. Cook, "Derivation of new banded waveguide model topology for sound synthesis," *Journal of the Acoustical Society of America*, vol. 133, no. 2, pp. EL76–EL81, 2013.

[10] A. Fink and A. Spanias, "Constrained estimation of percussive sound excitations," in *Proceedings of the IEEE Wokshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, pp. 201–204.

[11] ——, "Estimation of signal subspace-constrained inputs to linear systems," in *Proc. 45th Asilomar Conf. on Signals, Systems, and Computers*, 2011, to Appear.

[12] A. Fink, B. Mechtley, G. Wichern, J. Liu, H. Thornburg, A. Spanias, and G. Coleman, "Re-sonification of geographic sound activity using acoustic, semantic, and social information," in *Proceedings of the International Conference on Auditory Display (ICAD)*, Washington, D.C., June 2010.

[13] J. M. Kates, "Signal processing for hearing aids," in *Applications of digital signal processing to audio and acoustics*, M. Kahrs and K. Brandenburg, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 1998, pp. 235–277.

[14] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. A K Peters, Ltd., 2002.

[15] A. Valle, V. Lombardo, and M. Schirosa, "A graph-based system for the dynamic generation of soundscapes," in *Proceedings of the 15th International Conference on Auditory Display*, Copenhagen, 2009, pp. 217–224.

[16] A. Valle, M. Schirosa, and V. Lombardo, "A framework for soundscape analysis and re-synthesis," in *Proceedings of the Sound and Music Computing Conference*, Porto, Portugal, 2009.

[17] I. McGregor, A. Crerar, D. Benyon, and C. Macaulay, "Soundfields and soundscapes: Reifying auditory communities," in *Proceedings of the 2002 International Conference on Auditory Display*, Kyoto, 2002.

[18] G. Wichern, "Robust segmentation and retrieval of environmental sounds," PhD dissertation, Arizona State University, 2010.

[19] J. O. Smith III, *Spectral Audio Signal Processing*. W3K.

[20] M. Aramaki, "Analyse-synthèse de sons impulsifs: approches physique et perceptive," PhD dissertation, Université de la Méditerranée - Aix-Marseille II, 2002.

[21] M. Aramaki and R. Kronland-Martinet, "Analysis-synthesis of impact sounds by real-time dynamic filtering," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, pp. 695–705, March 2006.

[22] N. Lee and J. O. Smith III, "Use of the energy decay relief (edr) to estimate partial-overtone decay-times in a freely vibrating string," in *Program of the 152nd Meeting of the Acoustical Society of America*, Honolulu, Hawaii, USA, December 2006.

[23] R. Casati and A. Varzi, "Events," in *The Stanford Encyclopedia of Philosophy*, spring 2010 ed., E. N. Zalta, Ed., 2010.

[24] T. Hermann, A. Hunt, and J. G. Neuhoff, "Introduction," in *The Sonification Handbook*, T. Hermann, A. Hunt, and J. G. Neuhoff, Eds. Berlin, Germany: Logos Verlag, 2011, pp. 1–6.

[25] G. Kramer, B. Walker, T. Bonebright, P. R. Cook, J. Flowers, N. Miner, J. G. Neuhoff, R. Bargar, S. Barrass, J. Berger, G. Evreinov, W. T. Fitch, M. Gröhn, S. Handel, H. Kaper, H. Levkowitz, S. Lodha, B. Shinn-Cunningham, M. Simoni, and S. Tipei, "Sonification report: Status of the field and research agenda," International Community for Auditory Display, Tech. Rep., 1999.

[26] H. Thornburg and R. Leistikow, "An iterative filterbank approach for extracting sinusoidal parameters from quasi-harmonic sounds," in *Proc. IEEE WASPAA*, New Paltz, NY, 2003.

[27] ——, "Analysis and resynthesis of quasi-harmonic sounds: an iterative filterbank approach," in *Proc. of the 6th International Conference on Digital Audio Effects (DAFx-03)*, London, 2003.

[28] S. Ystad, P. Guillemain, and R. Kronland-Martinet, "Sound modeling of transient and sustained musical sounds," in *Proceedings of the International Computer Music conference (ICMC99)*, Beijing, China, October 1999, pp. 112–116.

[29] ——, "Sound modeling from the analysis of real sounds," in *Proceedings of the First International Conference on Digital Audio Effects (DAFX98)*, Barcelona, Spain, November 1998.

[30] K. Steiglitz, *A DSP Primer*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.

[31] J.-M. Adrien, "The missing link: modal synthesis," in *Representations of Musical Signals*, G. D. Poli, A. Piccialli, and C. Roads, Eds. Cambridge, MA, USA: MIT Press, 1991, pp. 269–298.

[32] R. Shepard, “Pitch perception and measurement,” in *Music, Cognition, and Computerized Sound*, P. R. Cook, Ed. Cambridge, MA, USA: MIT Press, 1999, pp. 149–165.

[33] P. R. Cook, Ed., *Music, Cognition, and Computerized Sound*. Cambridge, MA, USA: MIT Press, 1999.

[34] C. Roads, *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press, 1996.

[35] G. D. Poli, A. Piccialli, and C. Roads, Eds., *Representations of Musical Signals*. Cambridge, MA: MIT Press, 1991.

[36] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, “Discrete-time modelling of musical instruments,” *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, 2006.

[37] T. Tolonen, V. Välimäki, and M. Karjalainen, “Evaluation of modern sound synthesis methods,” Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Tech. Rep. 48, Mar. 1998.

[38] T. Painter and A. Spanias, “Perceptual coding of digital audio,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–515, Apr 2000.

[39] S. N. Levine, “Audio representations for data compression and compressed domain processing,” PhD dissertation, Stanford University, 1998.

[40] T. S. Verma, “A perceptually based audio signal model with application to scalable audio compression,” PhD dissertation, Stanford University, 2000.

[41] ISO/IEC 11172-3:1993, “Information technology âĂŤ coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s âĂŤ Part 3: Audio,” 1993, iSO.

[42] D. Rocchesso and F. Fontana, Eds., *The Sounding Object*. Online: http://www.soundobject.org/SObBook/, 2003 (Accessed 31 October 2011).

[43] M. V. Matthews, “The ear and how it works,” in *Music, Cognition, and Computerized Sound*, P. R. Cook, Ed. Cambridge, MA, USA: MIT Press, 1999, pp. 1–10.

[44] M. Matthews, "What is loudness?" in *Music, Cognition, and Computerized Sound*, P. R. Cook, Ed. Cambridge, MA, USA: MIT Press, 1999, pp. 71–78.

[45] A. Bregman, "Auditory scene analysis: The perceptual organization of sound," 1990.

[46] P. Oliveros, *Deep Listening: A Composer's Sound Practice*. iUniverse, 2005.

[47] R. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Rochester, VT: Destiny Books, 1977.

[48] J. Uys, "The gods must be crazy," 1980.

[49] A. Celentano, "Prisencolinensinainciusol," 1972.

[50] D. van Valkenburg and M. Kubovy, "From Gibson's fire to Gestalts: A bridge-building theory of perceptual objecthood," in *Ecological Psychoacoustics*, J. G. Neuhoff, Ed. San Diego, CA: Elsevier Science, 2004, pp. 113–147.

[51] L. D. Rosenblum, "Perceiving articulatory events: Lessons for an ecological psychoacoustics," in *Ecological Psychoacoustics*, J. G. Neuhoff, Ed. San Diego, CA: Elsevier Science, 2004, pp. 219–248.

[52] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[53] B. Cox, "Dr. Brian Cox tries to explain a gravity wave," Online Video Clip. YouTube, Jan. 2008, http://www.youtube.com/watch?v=6EJ8r9Rhfw8.

[54] L. Cremer, M. Heckl, and E. E. Ungar, *Structure-Borne Sound*, 2nd ed. Springer Verlag, 1988.

[55] H. Georgi, *The Physics of Waves, June 2007 Version*. http://www.people.fas.harvard.edu/ ~ hgeorgi/new.htm, Accessed 31 October 2011, online Book.

[56] J. d'Alembert, "Investigations on the curve formed by a vibrating string," in *Acoustics: Historical and Philosophical Development*, R. B. Lindsay, Ed. Dowden, Hutchinson & Ross, 1973, pp. 119–123.

[57] P. M. Morse and K. U. Ingard, *Theoretical Acoustics*. McGraw-Hill, 1968.

[58] B. Bank, "Physics-based sound synthesis of string instruments including geometric nonlinearities," PhD dissertation, Budapest University of Technology and Economics, 2006.

[59] A. H. Benade, *Fundamentals of Musical Acoustics*. New York, NY, USA: Oxford University Press, 1976.

[60] P. M. Ruiz, "A technique for simulating the vibrations of strings with a digital computer," PhD thesis, Music Master Dissertation, University of Illinois, Champaign-Urbana, Ill., 1969.

[61] A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. I. a physical model for a struck string using finite difference methods," *Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, February 1994.

[62] ——, "Numerical simulations of piano strings. II. comparisons with measurements and systematic exploration of some hammer-string parameters," *Journal of the Acoustical Society of America*, vol. 95, no. 3, pp. 1631–1640, March 1994.

[63] F. G. Friedlander, "On the oscillations of the bowed string," *Proc. of the Cambridge Philosophy Society*, vol. 49, pp. 516–530, 1953.

[64] J. B. Keller, "Bowing of violin strings," *Comm. Pure Applied Math.*, vol. 6, pp. 483–495, 1953.

[65] M. E. McIntyre and J. Woodhouse, "On the fundamentals of bowed-string dynamics," *Acustica*, vol. 43, pp. 93–108, 1979.

[66] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse, "On the oscillations of musical instruments," *J. Acoustic. Soc. of America*, vol. 74, no. 5, pp. 1325–1345, Nov. 1983.

[67] J. O. Smith III, "Efficient simulation of the reed-bore and bow-string mechanisms," in *Proc. of the ICMC*, 1986, pp. 275–280.

[68] ——, "Techniques for digital filter design and system identification with application to the violin," PhD dissertation, Stanford University, 1983.

[69] S. Serafin, "Sound design to enhance presence in photorealistic virtual reality," in *Proceedings of the 2004 International Conference on Auditory Display*, Sydney, 2004.

[70] G. Essl and P. R. Cook, "The principle of closed wavetrains, resonance and efficiency: past, present and future," in *Proceedings of the Stockholm Music Acoustics Conference*, September 2003, pp. 385–388.

[71] D. Rocchesso, "The ball within the box: A sound-processing metaphor," *Computer Music Journal*, vol. 19, no. 4, pp. 47–57, 1995.

[72] G. Essl, S. Serafin, P. R. Cook, and J. O. Smith III, "Musical applications of banded waveguides," *Computer Music Journal*, vol. 28, no. 1, pp. 51–63, 2004.

[73] H. L. F. von Helmholtz, *On the sensations of tone as a physiological basis for the theory of music*, second english ed. Longmans, Green, and Co., 1885.

[74] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, April 1965.

[75] J. O. Smith III and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proc. of the ICMC*, 1987.

[76] J. Laroche, "Using resonant filters for the synthesis of time-varying sinusoids," in *Proceeding of the 105th AES Convention*, San Francisco, CA, USA, 1998.

[77] ——, "Time and pitch scale modification of audio signals," in *Applications of digital signal processing to audio and acoustics*, M. Kahrs and K. Brandenburg, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 1998, pp. 279–310.

[78] C. Roads, *Microsound*. Cambridge, MA, USA: MIT Press, 2001.

[79] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," PhD dissertation, Stanford University, 1989.

[80] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[81] J. Laroche and M. Dolson, "Improved phase vocoder time-scale of audio," *IEEE Trans. on Audio and Speech Processing*, vol. 7, no. 3, pp. 323–332, May 1999.

[82] ——, "New phase-vocoder techniques for pitch-shifting, chorusing, harmonizing, and other wild effects," *J. Audio Eng. Soc.*, vol. 47, no. 11, pp. 928–936, Nov. 1999.

[83] X. Serra and J. O. Smith III, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[84] L. Almeida and F. Silva, "Variable-frequency synthesis: An improved harmonic coding scheme," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, Mar. 1984, pp. 437–440.

[85] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34, 1986.

[86] A. Röbel, "Parameter estimation for linear AM/FM sinusoids using frequency domain demodulation," in *Proceedings of the International Conference on Image and Signal Processing (ICISP)*, 2007.

[87] ——, "Frequency-slope estimation and its application to parameter estimation for non-stationary sinusoids," *Computer Music Journal*, vol. 32, no. 2, pp. 68–79, 2008.

[88] J. Nielsen, M. Christensen, A. Cemgil, S. Godsill, and S. Jensen, "Bayesian interpolation and parameter estimation in a dynamic sinusoidal model," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 1986–1998, Sept. 2011.

[89] S. N. Levine and J. O. Smith III, "A compact and malleable sines+transients+noise model for sound," in *Analysis, Synthesis, and Perception of Musical Sounds*, J. W. Beauchamp, Ed. Springer, 2007, pp. 145–174.

[90] T. S. Verma and T. H. Y. Meng, "Extending spectral modeling synthesis with transient modeling synthesis," *Computer Music Journal*, vol. 24, no. 2, pp. 47–59, 2000.

[91] H. Thornburg, "Detection and modeling of transient audio signals with prior information," Ph.D. dissertation, Stanford University, 2005.

[92] T. S. Verma, S. N. Levine, and T. H. Y. Meng, "Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals," in *Proceedings of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece, 1997.

[93] S. A. Van Duyne, J. R. Pierce, and J. O. Smith III, "Traveling wave implementation of a lossless mode-coupling filter and the wave digital hammer," in *Proc. of the ICMC*, 1994, pp. 411–418.

[94] D. A. Jaffe, "Silicon valley breakdown," Compact disc, 1994.

[95] P. R. Cook, "Identification of control parameters in an articulatory vocal tract model with applications to the synthesis of singing," Ph.D. dissertation, Stanford University, 1991.

[96] ——, "Singing voice synthesis: History, current work, and future directions," *Computer Music Journal*, vol. 20, no. 3, pp. 38–46, 1996.

[97] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *Journal of the Audio Engineering Society*, vol. 19, no. 6, pp. 462–470, 1971.

[98] ——, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part II," *Journal of the Audio Engineering Society*, vol. 19, no. 7, pp. 542–551, 1971.

[99] S. Bilbao, "Wave and scattering methods for the numerical integration of partial differential equations," PhD thesis, Stanford University, 2001.

[100] ——, *Wave and Scattering Methods for Numerical Simulation*. Chichester, UK: John Wiley and Sons, 2004.

[101] R. Rabenstein and L. Trautmann, "Digital sound synthesis of string instruments with the functional transformation method," *Signal Processing*, vol. 83, no. 8, pp. 1673–1688, August 2003.

[102] R. Marogna and F. Avanzini, "Physically-based synthesis of nonlinear circular membranes," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.

[103] R. Rabenstein, T. Koch, and C. Popp, "Tubular bells: A physical and algorithmic model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 881–890, 2010.

[104] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[105] D. A. Jaffe and J. O. Smith III, "Extensions of the Karplus-Strong plucked-string algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, 1983.

[106] J. O. Smith III, "A new approach to digital reverberation using closed waveguide networks," in *Proc. of the ICMC*, 1985, pp. 47–53.

[107] J. Rauhala and V. Välimäki, "Dispersion modeling in waveguide piano synthesis using tunable allpass filters," in *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Canada, 2006, pp. 71–76.

[108] M. Karjalainen, J. Backman, and J. Pölkki, "Analysis, modeling, and real-time sound synthesis of the kantele, a traditional finnish string instrument," in *Proceedings of the 1993 IEEE international conference on Acoustics, speech, and signal processing*, ser. ICASSP'93, 1993.

[109] S. Serafin, J. O. Smith III, and J. Woodhouse, "An investigation of the impact of torsion waves and friction characteristics on the playability of virtual bowed strings," ser. Proceedings of the IEEE Wokshop on the Applications of Signal Processing to Audio and Acoustic (WASPAA), New Paltz, NY, October 1999.

[110] B. Bank and L. Sujbert, "A piano model including longitudinal string vibrations," in *Proceedings of the International Conference of Digital Audio Effeccts (DAFx'04)*, Naples, Italy, 2004, pp. 89–94.

[111] ——, "Generation of longitudinal vibrations in piano strings: From physics to sound synthesis," *Journal of the Acoustical Society of America*, vol. 117, no. 4, pp. 2268–2278, April 2005.

[112] S. A. Van Duyne and J. O. Smith III, "Physical modeling with the 2-d digital waveguide mesh," in *Proc. of the ICMC*, 1993, pp. 40–47.

[113] ——, "The 2-d digital waveguide mesh," in *Proc. of the IEEE Wokshop on the Applications of Signal Processing to Audio and Acoustic (WASPAA)*, New Paltz, NY, USA, 1993.

[114] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *Journal of the Audio Engineering Society*, vol. 44, no. 5, pp. 331–353, 1996.

[115] V. Välimäki and T. Tolonen, "Development and calibration of a guitar synthesizer," *Journal of the Audio Engineering Society*, vol. 46, no. 9, pp. 766–778, 1998.

[116] J. O. Smith III, "Viewpoints on the history of digital synthesis," in *Proc. of the International Computer Music Conference ICMC*, Oct. 1991, pp. 1–10.

[117] K. van den Doel and D. K. Pai, "The sounds of physical shapes," *Presence*, vol. 7, no. 4, pp. 382–395, 1998.

[118] K. van den Doel, P. G. Kry, and D. K. Pai, "FoleyAutomatic: Physically-based sound effects for interactive simulation and animation," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM Press, 2001, pp. 537–544.

[119] K. van den Doel and D. K. Pai, "Modal synthesis for vibrating objects," in *Audio Anecdotes III: Tools, Tips, and Techniques for Digital Audio*. A. K. Peters, 2006, pp. 99–120.

[120] J. Laroche, "Etude d'un systeme d'analyse et de synthese utilisant la methode de prony. application aux instruments de musique de type percussif," PhD dissertation, ENST, 1989.

[121] K. Steiglitz and L. E. McBride, "A technique for the identification of linear systems," *IEEE Trans. Automat. Contr.*, vol. AC-10, 1965.

[122] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*. Prentice Hall, 1999.

[123] J. O. Smith III, *Introduction to Digital Filters with Audio Applications*. W3K Publishing, 2007.

[124] P. R. Cook and G. P. Scavone, "The synthesis toolkit (stk)," in *Proc. of the ICMC*, 1999, pp. 164–166.

[125] U. Uzmen, "Çizim tabletleri İçin fiziksel modelleme temelli sentezleyici (tr.: Physical modeling-based synthesizer employing a drawing tablet)," Master's thesis, Yıldız Teknik Üniversitesi, 2008.

[126] K. Steiglitz, "A note on constant-gain digital resonators," *Computer Music Journal*, vol. 18, no. 4, pp. 8–10, 1994.

[127] A. Misra, P. R. Cook, and G. Wang, "Musical tapestry: Re-composing natural sounds," in *Proceedings of the International Computer Music Conference (ICMC)*, New Orleans, LA, USA, 2006.

[128] C. Verron, "Synthèse immersive de sons d'environment," PhD dissertation, l'Universite Aix-Marseille I, 2010.

[129] D. Birchfield, N. Mattar, and H. Sundaram, "Design of a generative model for soundscape creation," in *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[130] P. Cano, L. Fabig, F. Gouyon, M. Koppenberger, A. Loscos, and A. Barbosa, "Semi-automatic ambiance generation," in *Proceedings of the International Conference of Digital Audio Effeccts (DAFxï£¡04)*, Naples, Italy, 2004.

[131] B. Truax, *Acoustic Communication*. Norwood, NJ: Ablex Publishing, 2001.

[132] Google Inc., "Corporate Information - Technology Overview," http://www.google.com/corporate/tech.html.

[133] G. Wichern, H. Thornburg, and A. Spanias, "Unifying semantic and content-based approaches for retrieval of environmental sounds," in *Proceedings of the IEEE Wokshop on the Applications of Signal Processing to Audio and Acoustic (WASPAA)*, New Paltz, NY, 2009.

[134] J. Kruskal and M. Wish, *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications, 1978.

[135] A. Valle and V. Lombardo, "A two-level method to control granular synthesis," in *Proc. of the 14-th Colloquium on Musical Informatics*, 2003, pp. 136–140.

[136] D. Rocchesso, R. Bresin, and M. Fernström, "Sounding objects," *IEEE MultiMedia*, vol. 10, no. 2, pp. 42–52, 2003.

[137] M. Aramaki, M. Besson, R. Kronland-Martinet, and S. Ystad, "Controlling the perceived material in an impact sound synthesizer," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 301–314, Feb. 2011.

[138] J. Laroche and J.-L. Meillier, "Multichannel excitation/filter modeling of percussive sounds with application to the piano," *IEEE Trans. on Speech and Audio Processing*, vol. 2, pp. 329–344, 1994.

[139] J. E. Bower, S. Lindroth, J. Burchett, S. D. Feller, D. J. Brady, and R. Brady, "soundSense: Sonifying pyroelectic sensor data for an interactive media event," in *Proceedings of the International Conference on Auditory Display*, July 2005, pp. 113–120.

[140] A. Constantinides, "Frequency transformations for digital filters," in *Electronics Letters*, vol. 3, 1967, pp. 487–489.

[141] J. A. Moorer, "The manifold joys of conformal mapping: Applications to digital filtering in the studio," *Journal of the Audio Engineering Society*, vol. 31, no. 11, pp. 826–841, 1983.

[142] S.-C. Pei and C.-C. Tseng, "IIR multiple notch filter design based on allpass filter," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 44, no. 2, pp. 133–136, Feb. 1997.

[143] M. Lang and T. Laakso, "Simple and robust method for the design of allpass filters using least-squares phase error criterion," in *IEEE Trans. on Circ. and Sys. II: Analog and DSP*, vol. 41, Jan. 1994, pp. 40–48.

[144] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay: Tools for fractional delay filter design," *Signal Processing Magazine, IEEE*, vol. 13, no. 1, pp. 30–60, Jan. 1996.

[145] R. G. Lyons and A. Bell, "The Swiss army knife of digital networks," in *Streamlining Digital Signal Processing: A Tricks of the Trade Guidebook*, R. G. Lyons, Ed. IEEE Press, 2007, pp. 283–300.

[146] N. Lee and J. O. Smith III, "Low-order allpass interpolated delay loops," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.

[147] P. R. Cook and G. P. Scavone, "The synthesis toolkit in C++ (STK)," https://ccrma.stanford.edu/software/stk/, accessed 9 March 2013.

[148] S. Sinclair, G. Scavone, and M. M. Wanderley, "Audio-haptic interaction with the digital waveguide bowed string," in *Proceedings of the International Computer Music Conference*, 2009, pp. 275–278.

[149] L. Cremer, *The Physics of the Violin*. tr. by J. S. Allen, Cambridge, MA, USA: MIT Press, 1984.

[150] G. Essl, "The displaced bow and APhISMs: Abstract physically informed synthesis methods for composition and interactive performance," in *Proceedings of the Twelfth Annual Florida Electroacoustic Music Festival*, 2003.

[151] S. A. Van Duyne and J. O. Smith III, "Developments for the commuted piano," in *Proceedings of the International Computer Music Conference*, 1995, pp. 335–343.

[152] J. O. Smith III and S. A. Van Duyne, "Commuted piano synthesis," in *Proceedings of the International Computer Music Conference*, 1995, pp. 319–326.

[153] R. V. Migneco, "Analysis and synthesis of expressive guitar performance," PhD dissertation, Drexel University, 2012.

[154] M. Aramaki, C. Gondre, R. Kronland-Martinet, T. Voinier, and S. Ystad, "Thinking the sounds: An intuitive control of an impact sound synthesizer," in *Proc. of the 15th Int. Conf. on Auditory Display*, 2009.

[155] N. Lee, Z. Duan, and J. O. Smith III, "Excitation signal extraction for guitar tones," in *Proc. of the ICMC*, 2007, pp. 450–457.

[156] H. Lee and M.-J. Tahk, "Generalized input-estimation technique for tracking maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, October 1999.

[157] H. E. Rauch, F. F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[158] R. Gessing, "Recursive smoothing for discrete-time systems as a filtering problem," *Control Theory and Applications, IEE Proceedings D*, vol. 131, no. 4, pp. 140–141, July 1984.

[159] Y. Zhang, Q. Li, G. Dai, and H. Zhang, "A new recursive least-squares identification algorithm based on singular value decomposition," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 2, Dec 1994, pp. 1733 –1734 vol.2.

[160] A. Gray and J. Markel, "Distance measures for speech processing," *IEEE Trans. on Acoustics, Speech, and Signal Processing,*, vol. 24, no. 5, pp. 380 – 391, Oct 1976.

[161] [Online]. Available: http://www.public.asu.edu/ ~ amfink/waspaa2011/sounds.html, Accessed: July 2011.

[162] S. Gillijns and B. D. Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, pp. 111–116, 2007.

[163] ——, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, pp. 934–937, 2007.

[164] B. Bank, "Physics-based sound synthesis of the piano," Master's Thesis, Budapest University of Technology and Economics, 2000.

[165] J. Laroche and J.-M. Jot, "Analysis/synthesis of quasi-harmonic sounds by use of the karplus-strong algorithm," *Journal de Physique IV*, vol. 2, no. C1, pp. 117–120, April 1992.

[166] M. Aramaki, J. Bensa, L. Daudet, P. Guillemain, and R. Kronland-Martinet, "Resynthesis of coupled piano string vibrations based on physical modeling," *Journal of New Music Research*, vol. 30, no. 3, pp. 213–226, 2001.

[167] M. Karjalainen, V. Välimäki, and P. A. A. Esquef, "Efficient modeling and synthesis of bell-like sounds," in *Proceedings of the International Conference of Digital Audio Effeccts (DAFx-02)*, Hamburg, Germany, 2002, pp. 181–186.

[168] C. R. Sullivan, "Extending the Karplus-Strong algorithm to synthesize electric guitar timbres with distortion and feedback," *Computer Music Journal*, vol. 14, no. 3, pp. 26–37, 1990.

[169] J. O. Smith III, personal communication, April 2012.

[170] T. Tolonen and H. Järveläinen, "Perceptual study of decay parameters in plucked string synthesis," in *Proceedings of the 109th Convention of the Audio Engineering Society*, Los Angeles, CA, USA, 2000.

[171] J. O. Smith III, "J.O. Smith III comments on Sullivan Karplus-Strong article," *Computer Music Journal*, vol. 2, no. 15, pp. 10–11, 1991.

[172] V. Välimäki, "Discrete-time modeling of acoustic tubes using fractional delay filters," PhD dissertation, Helsinki University of Technology, 1995.

[173] A. Papoulis, *Signal Analysis*. New York, NY, USA: McGraw-Hill, 1977.

APPENDIX A

POLES AND RESONANCE IN DIGITAL PROPAGATION MODELS

This appendix serves to describe the behavior of poles and related spectral peaks, or *resonances*, in digital delay-based propagation models. Specifically, the departure of pole angles and resonant frequencies of digital delay-based propagation models from expected harmonically related values, due to modeling of losses, even when phase delay is constant across all frequencies, is examined. As discussed below, this deviation is often very minor, perhaps imperceptible, and sometimes considered desirable; however, as banded waveguides separately model the modes of a system, accuracy in tuning may be important in model comparison. Additionally, the ranging possibility of transformations of and nonlinear interactions with sounding object models warrants caution and consideration of this phenomenon when designing and analyzing such models. Though focused on the effects of loss-modeling elements, alteration of poles and models due to lossless, dispersive elements is also discussed.

The rest of this appendix is organized as follows. First, the tuning of partials in analog systems and models is briefly described. This is followed by an examination of poles and resonances in digital delay-based propagation models. The appendix concludes with a discussion of this behavior, as well as its importance and perceptibility.

A.1 Analog Systems and Models

Analog modeling of losses in strings is usually achieved by the introduction of additional terms to the one-dimensional wave equation to approximate the loss of energy to terminations, drag, sound radiation, and friction [2, 60]. Whereas even-order terms correspond to dispersion, odd-order terms are associated with losses, and only a few typically suffice for good approximation [2]. While these terms primarily model losses, and therefore the decay of partials, they also affect tuning [60, 101].

In analog string models, a first-order time derivative gives rise to frequency-independent loss (i.e., the same loss for all frequencies), and higher, odd-order derivatives

induce frequency-dependent losses. Considering this, Ruiz proposed a string model of the form (note that a fourth-order term to model dispersion has been omitted) [60]:

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - 2b_1 \frac{\partial y}{\partial t} + 2b_3 \frac{\partial^3 y}{\partial t^3}. \tag{A.1}$$

This model, however, is ill-posed, and its finite difference digitization is prone to instability, particularly at high sample rates [1, 2]. Therefore, (A.1) is commonly replaced by

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - 2b_1 \frac{\partial y}{\partial t} + 2b_2 \frac{\partial^3 y}{\partial x^2 \partial t}, \tag{A.2}$$

where a mixed-derivative term replaces the third-order time derivative [61]. The coefficients b_1, b_2, and b_3 may all be found experimentally or based on knowledge of the modeled system.

Given its use in more recently developed finite difference models [1, 111], we consider the analog model of (A.2). Where the string described by (A.2) is "hinged" [61] or "pinned" [1] at the ends ($x = 0, L$), such that the boundary conditions are

$$y(0, t) = y(L, t) = 0 \tag{A.3}$$

$$\frac{\partial^2 y}{\partial x^2}(0, t) = \frac{\partial^2 y}{\partial x^2}(L, t) = 0, \tag{A.4}$$

it will have modal frequencies (i.e., the imaginary component of the system's poles) of the form

$$\omega_n = \sqrt{-\left(b_1 + b_2 \beta_n^2\right)^2 + c^2 \beta_n^2}, \tag{A.5}$$

where

$$\beta_n = \frac{n\pi}{L}, \tag{A.6}$$

with associated exponential decay rates

$$\sigma_n = b_1 + b_2 \beta_n^2 \tag{A.7}$$

Table A.1: Physical parameters of an example C4 string [1]

Parameter	Value	Units
c	329.6	ms
b_1	1.1	s^{-1}
b_2	2.7×10^{-4}	m^2/s
L	0.63	m

for all integers $n \neq 0$ [1]. From (A.7), it is clear that the first-order time derivative is associated with frequency-independent losses and the mixed derivative with frequency-dependent losses. The inclusion of either or both loss terms results in the modal frequencies being a nonlinear function of partial number, n, as defined by (A.5). Thus, the described string will have inharmonic modal frequencies; however, the deviation from the harmonic modes of a similar, lossless string is usually minute.

Consider a C4 piano string, such as the one given in [1], with the physical parameters listed in Table A.1. (Note that stiffness is ignored.) The modal frequencies of this model increasingly stray from their harmonic counterparts as frequency increases, but the mistuning is less than 0.000110 cents from 0 to 22050 Hz. Even the 1000th partial at approximately 261.6 kHz, well beyond human hearing, is mistuned by only 0.0144 cents. If only frequency-independent losses associated with the first-order term are considered (i.e., $b_2 = 0$), the mistuning decreases as frequency increases. In this case, the fundamental is the most mistuned mode, being flat by 0.000387 cents.

As evidenced by this example, losses in analog system models cause a shift in modal frequencies, away from the exact harmonic frequencies. This shift, however, is very small and perhaps negligible, especially when compared to any frequency shift due to dispersion. For many practical purposes, we may consider analog string models and similar systems without stiffness or dispersion to exhibit harmonically related modes.

A.2 Digital Delay-Based Propagation Models

Digital delay-based propagation models include digital waveguide (DWG) models [2], Karplus-Strong (KS) models [104, 105], single delay-loop (SDL) models [4], and banded waveguide (BWG) models [6, 7] . As described in Chapter 2, these are all digital systems that model wave propagation through the use of digital delays in combination with other elements. With such systems treated as a digital filter, with points of input and output, the denominator of the transfer function corresponds to a *string loop* with all model elements lumped in the feedback path (assuming the model is composed of linear elements) [4]. The string loop determines the poles, and hence the modal frequencies, of the model.

When a string loop is composed of integer delays and a scalar gain g, such that $0 < g \leq 1$, the poles will lie on a circle about the origin, with harmonically related angles. Thus, the impulse response of the system (a feedback comb filter) will be a sum of exponentially decaying sinusoids at perfectly harmonic frequencies (including DC), and the frequency response will have resonances at these same frequencies. This is expected as all frequencies are delayed by the same amount. Such a string loop corresponds to a model with frequency-independent loss and no dispersion.

Frequency-dependent loss may be modeled by introduction of a frequency-dependent filter, such that round-trip loss varies as a function of frequency. Figure A.1 shows a string loop of this type, with the transfer function

$$S(z) = \frac{1}{1 - G(z)z^{-L}}. \tag{A.8}$$

Frequency-dependent phase delay or group delay may be averted by use of zero-phase filters (or linear-phase filters with suitably adjusted delay lines) [2, 123], so as to avoid the dispersive effects of such on partial frequencies. The pole angles and resonant frequencies,

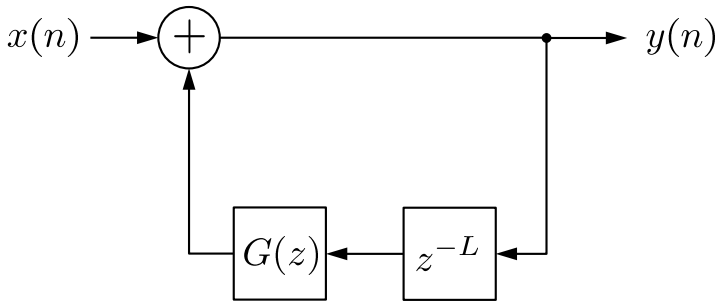


Figure A.1: An example string loop with integer delay and a zero-phase loss filter.

however, of such systems are affected by the use of a loss filter. This phenomenon has been recognized by many [2, 30, 164–166], but overlooked at times [8, 167, 168]

It is important to note that in digital delay-based propagation models, the resonant frequencies are not necessarily the same as the system's pole angles (i.e., the modal frequencies or eigenfrequencies). This is a feature common to many types of digital filters, such as relatively simple resonant biquad filters [30, 123, 126]. The difference between pole angles and resonant frequencies results from the combined effect of all of a system's roots. In a simple second-order autoregressive filter with complex-conjugate poles, for example, the "skirts" of the resonances (one of which is at a negative frequency) caused by the two poles interfere with one another [30, 123], shifting the spectral peak. In general, this disparity is lessened in systems when poles are moved closer to the unit circle and further away from one another.

A.2.1 Resonances

The frequency response of a system as in Figure A.1, with $L = 32$ and $G(z) = 0.95[0.24z^1 + 0.52 + 0.24z^{-1}]$, is shown in Figure A.2, with dashed lines indicating the expected harmonic frequencies (i.e., the frequency $\frac{2\pi}{L}$ and its subsequent harmonics). Though the effect is quite slight, it may be seen that the peaks, especially those at the

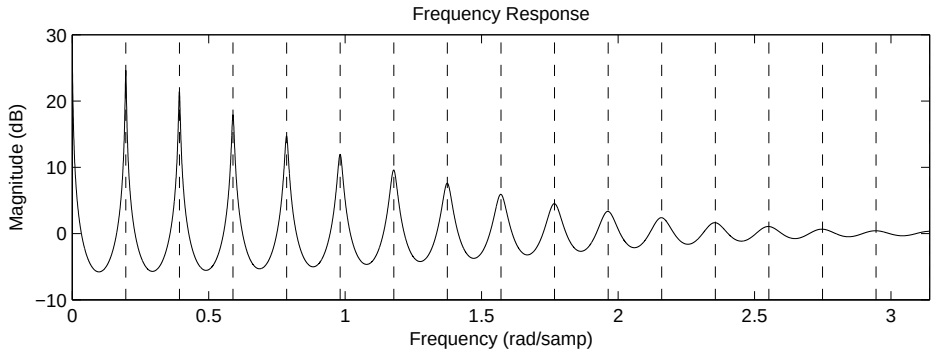


Figure A.2: Frequency response of a string loop model with a linear-phase loss filter.

more damped, higher frequencies do not exactly align with the expected harmonic frequencies. Modeled after a system in [30], this simple example with a second-order loss filter illustrates the inharmonicity of resonant peaks in propagation models. Following the principle of wave train closure [54, 70], one might expect the resonant frequencies in such propagation models to occur at the expected harmonic frequencies, where traveling waves meet themselves in-phase and interfere constructively. However, when frequency-dependent losses are low-pass in nature, the lesser attenuation at a lower-than-harmonic frequency combined with its almost maximal self-interference may be greater than the combination of more attenuation and purely constructive interference at the nearby harmonic frequency.

The emergence of inharmonically-related resonant frequencies in propagation models with constant phase delay has been noted in previous works. In [30], Steiglitz illustrates that in a Karplus-Strong model with a linear-phase loss filter, the resonances do not align with a perfectly harmonic progression, particularly for systems with a small phase delay. Additionally, in [166], the authors imply that in a DWG-type model with frequency-dependent gain modeled by a linear-phase filter, the resonant frequencies are not perfectly harmonic, but are nearly so, given that the round-trip gain of most fre-

quencies is near unity. (The implication arises from the more general treatment of loss-modeling filters with arbitrary phase response.) These mentions touch on the conditions for harmonically related resonances, explicitly given below. It follows from these conditions that for models where loss filters have a near-unity [166] or almost flat magnitude frequency response, the resonances are approximately harmonic.

Conditions for Harmonicity of Resonances

Regarding models of the form of that shown in Figure A.1, where $G(z)$ is a zero-phase filter, resonances will only align harmonically if at every expected harmonic frequency, the loss filter either has a gain of one or an instantaneously flat magnitude frequency response. To prove this, consider a generic string loop as in Figure A.1, with a frequency response of the form

$$S(e^{j\omega}) = \frac{1}{A_S(e^{j\omega})} = \frac{1}{1 - G(\omega)e^{-j\omega L}}, \tag{A.9}$$

where $G(\omega) \in \mathbb{R}$ is the DTFT of an arbitrary zero-phase filter such that $0 \leq G(\omega) \leq 1 \; \forall \; \omega \in \mathbb{R}$. The resonant frequencies (i.e., the spectral maxima) then occur at the minimal values of $|A_S(e^{j\omega})|$. To find the resonant frequencies, one may then note that the minima of $|A_S(e^{j\omega})|$ as a function of ω occur at the minima of

$$|A_S(e^{j\omega})|^2 = 1 - 2G(\omega)\cos(L\omega) + G^2(\omega). \tag{A.10}$$

These minimal points occur where

$$\frac{d(|A_S(e^{j\omega})|^2)}{d\omega} = 2\left[(G(\omega) - \cos(L\omega))\frac{dG(\omega)}{d\omega} + G(\omega)L\sin(L\omega)\right] = 0. \tag{A.11}$$

To determine if the resonant frequencies occur at the expected harmonic frequencies, one may see if the minima of (A.11) are at the frequencies

$$\omega_k = \frac{2\pi k}{L}, \tag{A.12}$$

where $k = 0, 1, \ldots, L$. Evaluating (A.11) at any ω_k, one obtains

$$\left.\frac{d(|A_S(e^{j\omega})|^2)}{d\omega}\right|_{\omega=\omega_k} = 2(G(\omega_k)-1)\left.\frac{dG(\omega)}{d\omega}\right|_{\omega=\omega_k}. \tag{A.13}$$

Clearly, (A.13) is only equal to zero if either

$$G(\omega_k) = 1 \tag{A.14}$$

or

$$\left.\frac{dG(\omega)}{d\omega}\right|_{\omega=\omega_k} = 0. \tag{A.15}$$

That is, the resonances of (A.9) are only at the harmonic frequencies, ω_k for $k = 0, 1, \ldots, L$, if the loss filter's frequency response is equal to unity or has a slope of zero at each ω_k. This condition is obviously not met for relatively low-order low pass filters.

Resonance in Models with Nonlinear-phase Elements

The above conditions for harmonicity of resonances in digital-delay based propagation models may easily be extended to conditions for which resonances occur at any frequency at which the propagation-modeling elements cause a phase shift of an integer multiple of 2π. That is, if a model as in (A.9) is altered by the introduction of nonlinear-phase elements in the propagation path, such as allpass filters for fractional delay or stiffness modeling, the resonances will occur at the expected frequencies of phase closure if $G(\omega)$ is equal to one or has a slope of zero at those frequencies.

A.2.2 Poles

The pole angles and radii of digital delay-based propagation models represent simulated modal frequencies and decay rates, respectively. In analog models with losses and without explicitly modeled dispersion, the modal frequencies are very nearly harmonically related (cf. Section A.1). In comparable digital models, a similar nearly harmonic distribution of poles angles is desirable (from a modeling standpoint). Though the filters used generally only approximately model losses [2, 114, 115], one may still expect

wave train closure [54, 70] to account for the modal frequencies. As mentioned above, however, this is not the case, even when modeling losses via linear-phase filters so as to have constant phase delay. This behavior of poles in propagation models has been acknowledged elsewhere, including applications to Karplus-Strong models [165], piano modeling [164], and reverberation [2].

This mistuning of modes is often slight in practice [2, 164, 165]. As with the resonant frequencies of a system, the inharmonicity of pole angles may also be lessened as round-trip losses decrease or sampling rate increases (assuming model parameters are appropriately adjused). This change in pole angles may be viewed as a result of using an "approximate conformal map" that does not preserve the linearity of lines that intersect the z-plane's origin [2], when modifying a lossless string loop by introduction of a loss filter. That is, (A.8) may arise from the transfer function,

$$S_u(z) = \frac{1}{1 - z^{-L}} \tag{A.16}$$

when making the substitution $z^{-1} \leftarrow G^{1/L}(z)z^{-1}$, which is not guaranteed to preserve pole angles [2]. However, when poles are near the unit circle and $G^{1/L}(z)$ is near unity along the unit circle, as is often the case in models of resonant musical objects, pole angles shift only slightly [2].

Pole Shift from Fractional Delay and Dispersion Modeling

In digital delay-based propagation models where the propagation distance is not equivalent to an integer number of samples at any frequency, fractional delay filters may be used to correct the phase delay of the propagation elements. Discussed further in Appendix B, such filters generally have nonlinear phase and may be implemented by allpass filters. By the principle of wave train closure, as with linear-phase models, one may expect the pole angles in such models to occur at frequencies for which the total phase response of all propagation modeling elements is an integer multiple of 2π. However, in

such models where there is loss (i.e., where the poles are not on the unit circle), the poles will not fall at the ideal locations.

Consider a simple string loop, with scalar loss and an allpass fractional delay filter, of the form

$$S_f(z) = \frac{1}{1 - g z^{-L_i} F(z)}, \tag{A.17}$$

where $0 < g < 1$, $L_i \in \mathbb{Z}^+$, and $F(z)$ is a real first-order allpass filter of the form

$$F(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}}. \tag{A.18}$$

By following typical DWG design methodology, to create an overtone at frequency ω_m that decays by R_m per sample, with a propagation length of k wavelengths, one would design a feedback loop with total phase delay $L_m = \frac{k2\pi}{\omega_m}$ and gain, $g = R_m^{L_m}$. Implementing the propagation delay as in (A.17), the integer delay will be of length L_i, and the phase delay of $F(z)$ at ω_m will be L_f, such that $L_m = L_i + L_f$. By evaluating (A.17) at the desired pole, $z_m = R_m e^{j\omega_m}$, one may see that it does not have a pole at this location on the z-plane. For this point to be a pole requires

$$z_m^{-Li} F(z_m) = R_m^{-L_m} e^{-j\omega_m L_m} = \frac{1}{g}, \tag{A.19}$$

which only holds if, in general,

$$F(z_m) = R_m^{-L_f} e^{-j\omega_m L_f}. \tag{A.20}$$

This condition is only met in general for $|z| < 1$ when $F(z) = z^{-L_f}$, which is not realizable in practice. Discussed more in Appendix B, a pole may be placed at the desired location by concurrent design of the gain, g, and the fractional delay filter.

Pole Analysis of a Simple System

To show that digital delay-based propagation models with constant phase delay do not always have poles at the expected harmonically-related angles (i.e., the angles that correspond to frequencies with an integer number of wavelengths equal to the phase delay),

it is shown here that a digital delay-based propagation model with a third-order linear-phase loss filter does not have poles at angles corresponding to the expected harmonic frequencies. Note that this discussion focuses on poles that do not lie on the real line of the z-plane, and it also does not address the added system poles due to the loss filter's increase of system order.

Consider a KS or SDL model with integer delay and only a single scalar multiply to represent frequency-independent loss with a transfer function of the form

$$H_1(z) = \frac{B_1(z)}{A_1(z)} = B_1(z) \cdot \left[1 - g z^{-L}\right]^{-1}, \tag{A.21}$$

where $B_1(z)$ represents some FIR transfer function (of order less than L), accounting for the points of interaction and observation. If $g = 1$, the poles of $H_1(z)$ will be the Lth roots of unity,

$$W_L^k = e^{-j\frac{2\pi}{L}k}, \quad k = 0, 1, 2, \ldots, L-1, \tag{A.22}$$

located along the unit circle on the z-plane, every $\frac{2\pi}{L}$ radians. Otherwise, for other values of g, the poles will be located at

$$g^{1/L} W_L^k, \quad k = 0, 1, 2, \ldots, L-1, \tag{A.23}$$

still located every $\frac{2\pi}{L}$ radians, but on a circle of radius $g^{1/L}$. These poles occurring at uniformly-spaced angles will correspond to sinusoids (damped for $g < 1$) at harmonically-related frequencies.

If a third-order zero-phase filter (or linear-phase filter for practical implementation) is used to model losses, the transfer function will be of the form

$$H_2(z) = \frac{B_2(z)}{A_{2(z)}} = B_2(z) \cdot \left[1 - g_1 z^{-(L-1)} - g_0 z^{-L} - g_1 z^{-(L+1)}\right]^{-1}. \tag{A.24}$$

For the pole angles to correspond to the expected harmonic frequencies, each pole must be of the form

$$z_k = R_k e^{-j\frac{2\pi}{L}k} = R_k W_L^k, \qquad k = 0, 1, \ldots, L-1, \tag{A.25}$$

where $0 \leq R_k \leq 1$. For each z_k to be a pole of (A.24), the denominator must satisfy the equation

$$A_2(z_k) = 1 - g_1 R_k^{-(L-1)} W_L^{-k(L-1)} - g_0 R_k^{-L} W_L^{-kL} - g_1 R_k^{-(L+1)} W_L^{-k(L+1)} = 0. \tag{A.26}$$

The expression for $A_2(z_k)$ may be simplified to

$$A_2(z_k) = 1 - g_1 R_k^{-(L-1)} W_L^{k} - g_0 R_k^{-L} - g_1 R_k^{-(L+1)} W_L^{-k}, \tag{A.27}$$

which may alternatively be expressed as

$$A_2(z_k) = 1 - g_0 R_k^{-L} - g_1 R_k^{-L} \cos\left(\tfrac{2\pi}{L} k\right)(R_k + R_k^{-1}) + j \cdot g_1 R_k^{-L} \sin\left(\tfrac{2\pi}{L} k\right)(R_k - R_k^{-1}). \tag{A.28}$$

Note here, that is a necessary (though not sufficient) condition that for (A.26) to be true, that

$$\mathrm{Im}\{A_2(z_k)\} = 0. \tag{A.29}$$

For this to hold, at frequencies other than DC and the Nyquist frequency, when g_1 is nonzero, each R_k must then satisfy

$$R_k - R_k^{-1} = 0, \tag{A.30}$$

or alternatively,

$$R_k = \pm 1. \tag{A.31}$$

Thus, any modal frequency may only possibly be at the expected harmonic frequency if its associated pole is on the unit circle, meaning it does not decay and the system is marginally stable. When this is true of all poles, the transfer function's denominator corresponds to a simple feedback comb filter with $g_1 = 0$ and $g_0 = 1$. The system's poles can also lie at the expected angles (on a circle about the z-plane's origin) for other values of g_0 when $g_1 = 0$, corresponding to frequency-independent losses.

To illustrate the loss filter's mistuning effect in a system of the form of (A.24) the loci of the poles are shown in Figures A.3 and A.4 for a system where $L = 32$ and the loss filter is varied such that g_0 is swept between the values $0.95 \cdot 0.999$ and $0.95 \cdot 0.51$, while g_1 satisfies

$$2g_1 + g_0 = 0.95. \tag{A.32}$$

More informally, one may consider the loss filter varied between $G(z) = 0.95(0.0005 + 0.999z^{-1} + 0.0005z^{-2})$ and $G(z) = 0.95(0.245 + 0.51z^{-1} + 0.245z^{-2})$. In each figure, the unit circle and lines corresponding to the expected harmonic pole angles are drawn as dashed lines. Additionally, the plotted loci shift in color (where available) from black to blue as g_0 is decreased from $0.95 \cdot 0.999$ to $0.95 \cdot 0.51$. The loci of all of the poles are shown in Figure A.3, while Figure A.4 focuses on the higher frequency poles which exhibit more mistuning.

A.3 Perception, Detection, and Modeling of Resonance and Modes

Since the mistuning of partials in propagation models increases as pole radii decrease, severe mistuning is generally associated with very fast-decaying modes. Such rapidly decaying components may be perceived as transients, rather than as quasi-harmonic components of some sound. Where model parameters are estimated from sound recordings, fast-decaying modes are generally difficult to identify, due to their high bandwidth and the presence of noise. In many data-driven physical modeling applications, such components may even be ignored or non-parametrically modeled, incorporated into excitation signals [2, 14]. Primarily affecting high frequencies and highly damped modes, the mistuning of partials is generally not considered detrimental and is sometimes regarded as pleasantly affecting the timbre of outputs [169]. Indeed, many models use nonlinear-phase loss filters, without particular regard for the mistuning of partials [2, 114].

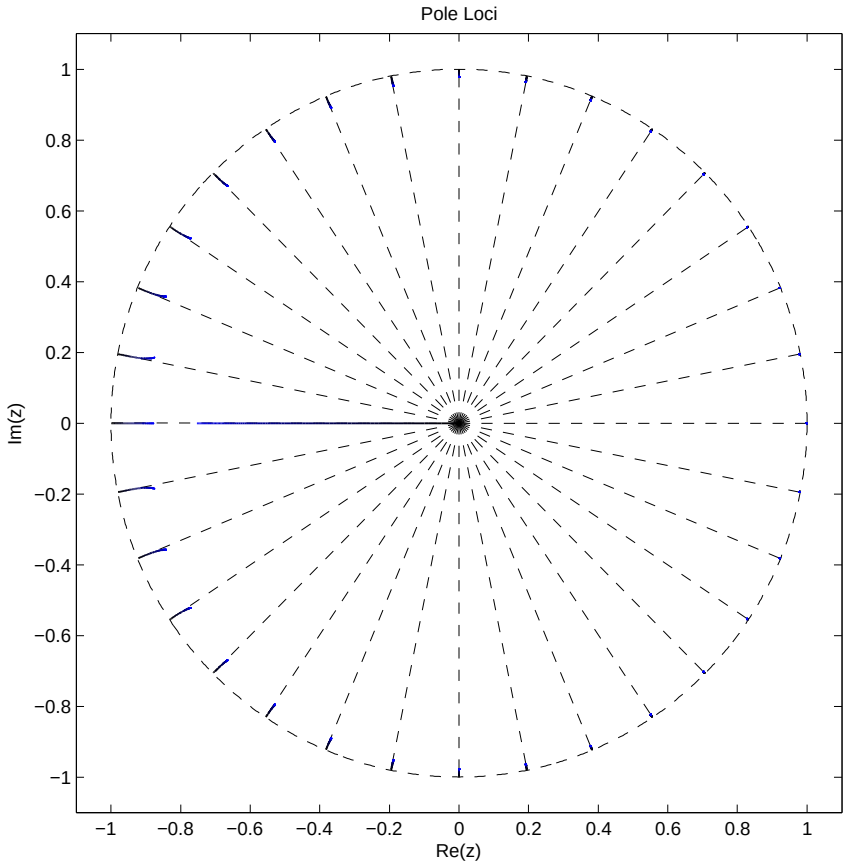


Figure A.3: Pole loci of a string loop as the linear-phase loss filter is varied.

In BWG models, with each mode separately modeled, it is much simpler to control the resonant and modal frequencies of each overtone. When comparing BWG models to other models, such as DWGs, designed to the same modal data or physical specifications, care should be taken to ensure correct tuning. In certain applications, perceptual significance may win over precision of modal properties [42, 170]. However, in applications with nonlinear interactions or the possibility for transforming objects, otherwise perceptually irrelevant modes may become perceptible or significantly affect simulation. In general, physical modeling should favor accuracy without application-specific justification for approximation.

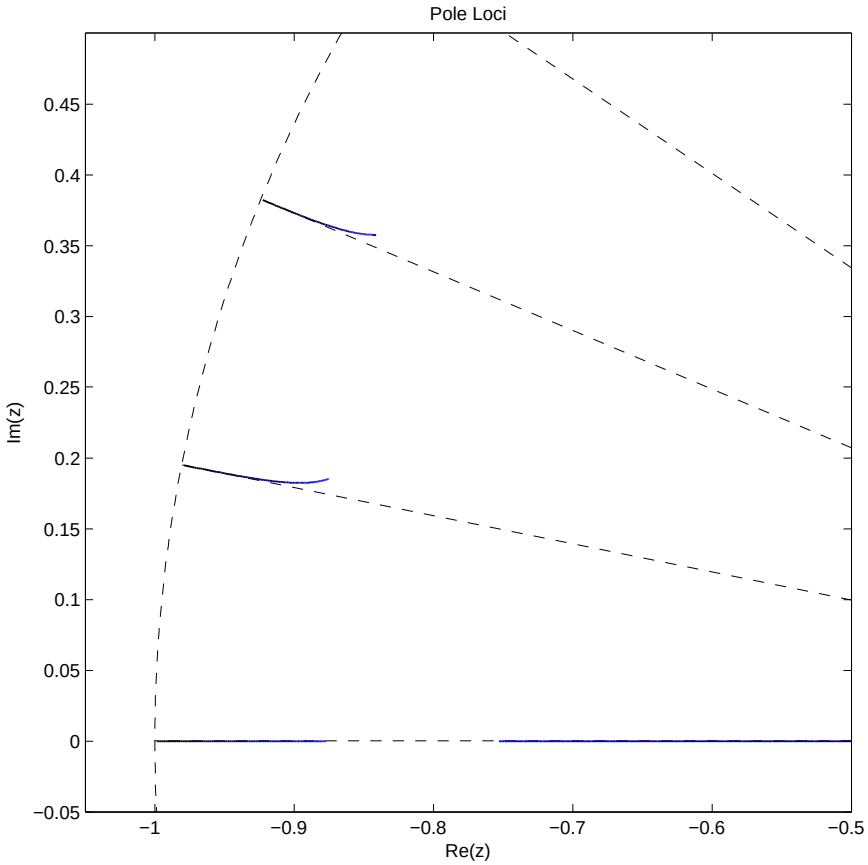


Figure A.4: Detail of high frequency pole loci of a string loop as the linear-phase loss filter is varied.

APPENDIX B

FRACTIONAL DELAY IN BANDED WAVEGUIDES

Fractional delay filters are an important feature of digital delay-based propagation models, allowing fine tuning of resonant and modal frequencies. With DWG models, for example, sole use of pure integer delays limits the possible fundamental frequencies of the model (when sampling rate is not varied), with mistuning being generally worse at higher frequencies [105]. Use of tunable fractional delay filters, however, allows finer tuning, as well as better simulation of dynamically varied string length (i.e., *glissandi*) [68, 171].

This appendix discusses the use of fractional delay filters in BWGs, where signals are *narrowband*, with energy focused within some frequency band. Specifically, the use of first-order allpass fractional delay filters is discussed. Such a filter is the minimally complex implementation of a LTI fractional delay filter that allows precise phase delay at the BWG's modal frequency, without leading to attenuation or instability . Additionally, the possibility of other types of fractional delay filters for use in BWGs is discussed. More general treatment of fractional delay filters may be found in [144], and fractional delay in the context of physical modeling is extensively described in [2, 172]

B.1 Summary of Fractional Delay Filtering

As a typical digital signal is limited to uniformly spaced samples, it may not be delayed as-is by an amount of time that is not a multiple of the sampling period. The ideal fractional delay filter, for non-integer delays, has an infinite and anti-causal transient response, corresponding to samples of an appropriately delayed continuous-time sinc function [144]. As such a filter is not realizable in practice, practical fractional delay filters require approximation of this ideal response or properties relating to it.

Common measures of fractional delay include group delay and phase delay. For a filter with a phase response, $\Theta(\omega)$, its group delay (measured in samples) is defined as

$$\tau_g(\omega) = -\frac{d\Theta(\omega)}{d\omega}. \tag{B.1}$$

At frequencies where the phase response is smooth, group delay is generally regarded as the amount by which the amplitude envelope of a sinusoid at the respective frequency is delayed [123, 173]. Alternatively, the phase delay of a filter, defined by

$$\tau_p(\omega) = -\frac{\Theta(\omega)}{\omega}, \tag{B.2}$$

may be interpreted as the time delay of a sinusoid at digital frequency ω [123]. Note that for a pure integer delay, the phase response is linear, leading to a group delay and phase delay that are equal to the integer delay amount. The same is true of linear-phase filters, excepting solely piecewise linear-phase filters [123].

Implementation of fractional delay filters is generally limited to linear interpolation (FIR) filters and allpass (IIR) filters, with the transient or phase response optimized in some sense, with reference to the ideal fractional delay filter [2, 144] or measures of fractional delay. Designs for FIR fractional delay filters abound, but such filters do not have a flat magnitude response, generally attenuating higher frequencies [144]. As BWG models focus on slowly decaying modes, with near-unity round-trip gain, such fractional delay filters are not generally appropriate for modeling propagation in BWGs. Overall fractional delay filter gains may be adjusted to correct attenuation at the frequency of interest; however, this may easily lead to instability. Such problems may be overcome by use of allpass fractional delay filters.

B.2 First-Order Allpass Fractional Delay Filters

For implementation of BWG models, this work considers IIR allpass fractional delay filters that, when set to unity gain, do not attenuate signal components of any frequency. Where a BWG uses a scalar gain to model losses, a first-order allpass filter allows precise tuning of the propagation-modeling filter's desired phase delay at the modal frequency without affecting attenuation or stability. This may be achieved by designing the allpass filter such that the total phase delay of the elements in the BWG's feedback loop

at the mode frequency is equal to an integer multiple of wavelengths. (Equivalently, the phase shift of propagation modeling elements should be a multiple of 2π at the mode frequency.) As discussed in Appendix A and below, this only results in approximate placement of a pole at the desired location on the z-plane, though the scalar gain and fractional delay filter could be designed simultaneously toward correct pole placement (in lieu of correct phase delay).

A first order allpass filter of the form

$$F(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}}, \tag{B.3}$$

with a pole at $z = -a_1$, may be designed to have phase delay, $\tau_p(\omega_m)$, at frequency ω_m by setting

$$a_1 = \frac{\sin\left(\omega_m \frac{1-\tau_p(\omega_m)}{2}\right)}{\sin\left(\omega_m \frac{1+\tau_p(\omega_m)}{2}\right)}. \tag{B.4}$$

At $\omega_m = 0$, the preceding equation has an undefined solution, but may be substituted by

$$a_1 = \frac{1 - \tau_p(\omega m)}{1 + \tau_p(\omega m)}, \tag{B.5}$$

which may also be used to approximate a_1 for relatively low frequencies [105]. Such a fractional delay filter has seen use for tuning the phase delay at the fundamental frequency of Extended Karplus-Strong models (though frequency dependent-losses will cause a *slight* mistuning, as discussed in the previous appendix) [105].

To examine this filter as a function of ω_m and $\tau_p(\omega_m)$, Figure B.1 shows the resulting pole angle as phase delay is varied between zero and one sample for all frequencies. Figure B.2 shows a similar plot, as phase delay is varied between zero and two samples; note that there is a region for higher frequencies and phase delays for which there is no real and stable first-order allpass filter to meet the design criteria. From these figures and inspection of (B.4), one may observe that for low values of phase delay, and for high digital frequencies of interest, the pole approaches $z = -1$.

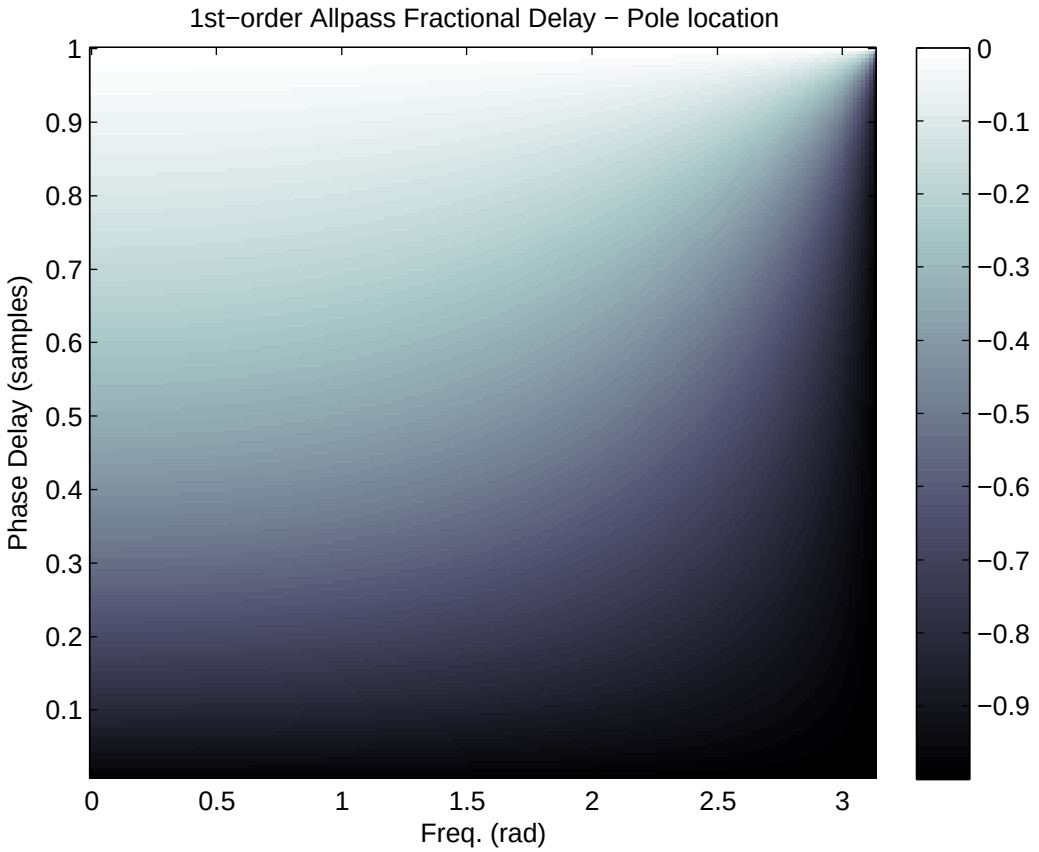


Figure B.1: Pole angle as a function of phase delay at a specified frequency for a first-order allpass fractional delay filter.

B.2.1 Issues in Implementation

As human audition tends to observe frequency logarithmically and many sounds exhibit low-pass characteristics, first-order allpass fractional delay filter design often uses the approximation of (B.5), to most accurately model fractional delay at low frequencies, even in tuning KS and DWG models [2]. Such an approximation may be used in any application, though the error in phase delay increases as a function of frequency.

As shown in Figures B.1 and B.2, low values of phase delay lead to a pole near the unit circle. If phase delay is set to zero, the allpass filter then has a pole on the unit circle, canceled by a zero with identical placement. As such cancellations are perilous

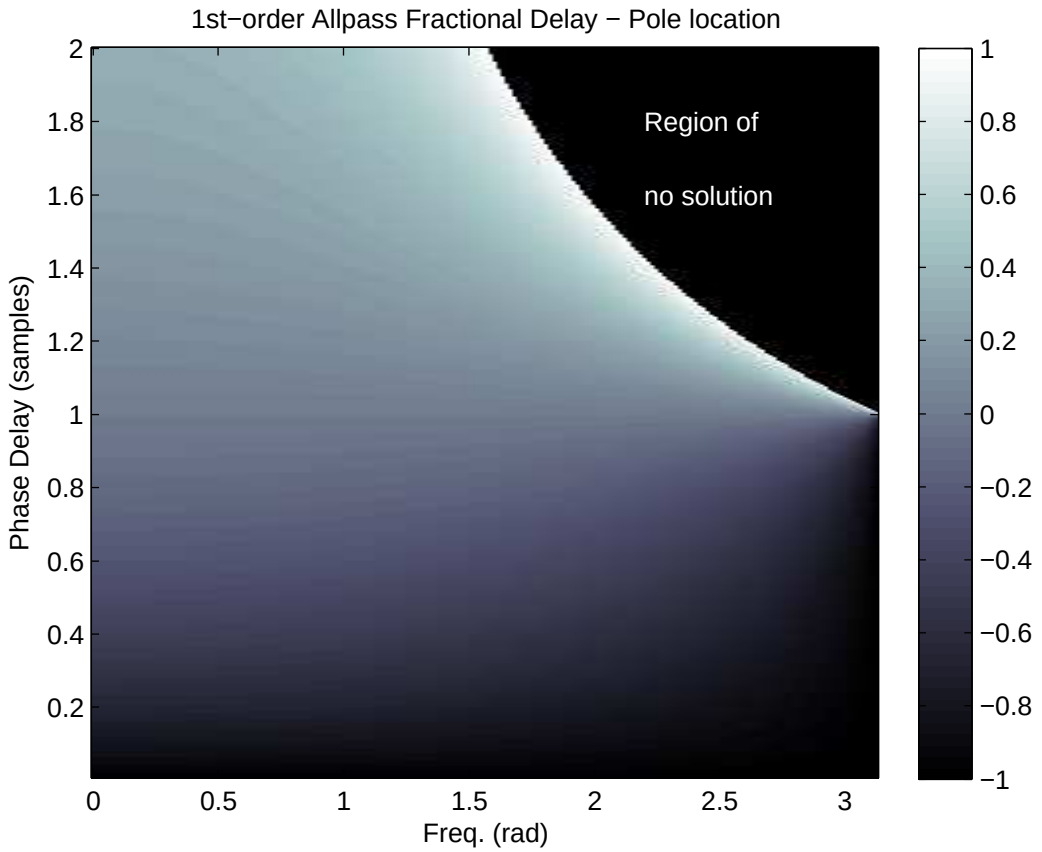


Figure B.2: Pole angle as a function of phase delay at a specified frequency for a first-order allpass fractional delay filter.. Note the region for which there is no real stable pole.

where there may be round-off error, they should generally be avoided. Thus, the range of phase delays for allpas fractional filter delay is often limited to some range $[\epsilon, 1+\epsilon]$, where $0 < \epsilon < 1$, disallowing zero delay [2]. Additionally, design may be limited by the fact that as the pole location approaches the unit circle, the sharp slope of the filter's phase response increases the group delay near the Nyquist frequency, causing the filter's transient response to display a high-frequency "ringing" that may not be desirable [2].

For narrowband applications, such as BWGs, one may ignore this ringing in many instances, as there will be little high-frequency energy in the system. Still, phase delay in first-order allpass fractional delay filters should be kept above zero, so as to prevent a

pole-zero cancellation. At high frequencies, with phase delay increasingly limited at the upper bound, pole placement near $z = -1$ becomes unavoidable. Thus, a measure of ringing may occur, depending on the details of the BWG implementation. However, in DWG-inspired BWG models, such as in Chapter 4, where the feedforward portion of the BPF and propagation-modeling feedback path are identically designed, any such ringing will effectively be canceled by interference. Though if these portions of the model do not implement the same fractional delay filters (as in a simplified bi-directional model), such cancellation may not occur.

B.2.2 Design for Pole Placement

Mentioned in Appendix A, designing for the correct phase delay in BWG models of the form given in this work (i.e., DWG-inspired BWG models) does not place a pole of the propagation-modeling feedback loop at the desired location. Precise pole placement may be achieved, however, by concurrent design of the round-trip gain and fractional delay filter. That is, to place a pole at $z_m = R_m e^{j\omega_m}$, one may solve

$$1 - g z_m^{-L_i} \frac{a_1 + z_m^{-1}}{1 + a_1 z_m^{-1}} = 0 \tag{B.6}$$

for g and $a1$, constrained by $0 < g < 1$ and $-1 < a_1 < 1$.

Design of the feedback loop in this manner increases the complexity of model design. For linear interactions at a point in BWG models configured as in Section 4.4, where a resonator (with the correct pole) is paired with inversely-related feedforward and feedback loops, the tuning of the pole in the feedback path will not even change the BWG's response, assuming there is no significant round-off error. Indeed, any arbitrary filter and its inverse could be used, since they cancel one another and the resonator's response determines the output of the BWG. In the presence of nonlinear feedback-based interactions though, variation of such details has potential to alter outputs.

In the simulations in this work, this design technique is not used, in the interest of computational parsimony. One may note that maintaining separate design of g and the fractional delay filter somewhat mirrors the typical separation of elements for tuning, loss, and dispersion in DWG modeling and related methods [2, 4, 105].

B.2.3 Further Properties and Other Fractional Delay Filters

Though specification of phase delay in a simple delayed feedback loop with frequency-independent loss allows precise pole angle placement, simple fractional delay filters may ignore components of propagation, such as group delay (and relatedly, group velocity). In designing first-order allpass fractional delay filters to specified phase delay at a frequency of interest, for example, the group delay at any frequency is solely a consequence of pole placement. To examine the resulting value of group delay in this type of filter, Figures B.3, B.4, and B.5 display the resulting group delay at the frequency of interest when specifying phase delay. Note that to better show the range of low values, Figures B.4 and B.5 limit the displayed range of group delay to a maximum value of two.

From these figures it may be seen that in using the simplest allpass fractional delay filter, when a BWG produces a high-frequency mode, the resulting group delay tends toward high values. (This may be viewed as a cause of the previously described high-frequency ringing.) This may not be problematic in some applications, especially if human perception is taken into account, but an inability to control various properties of modeled propagation limit the robustness of BWG models.

More complex fractional delay filters may of course be used in BWG models, designed to have specified phase delay, group delay, or other properties. As these filters increase in complexity however, they may become computationally prohibitive to implement or design, particularly for real time applications.

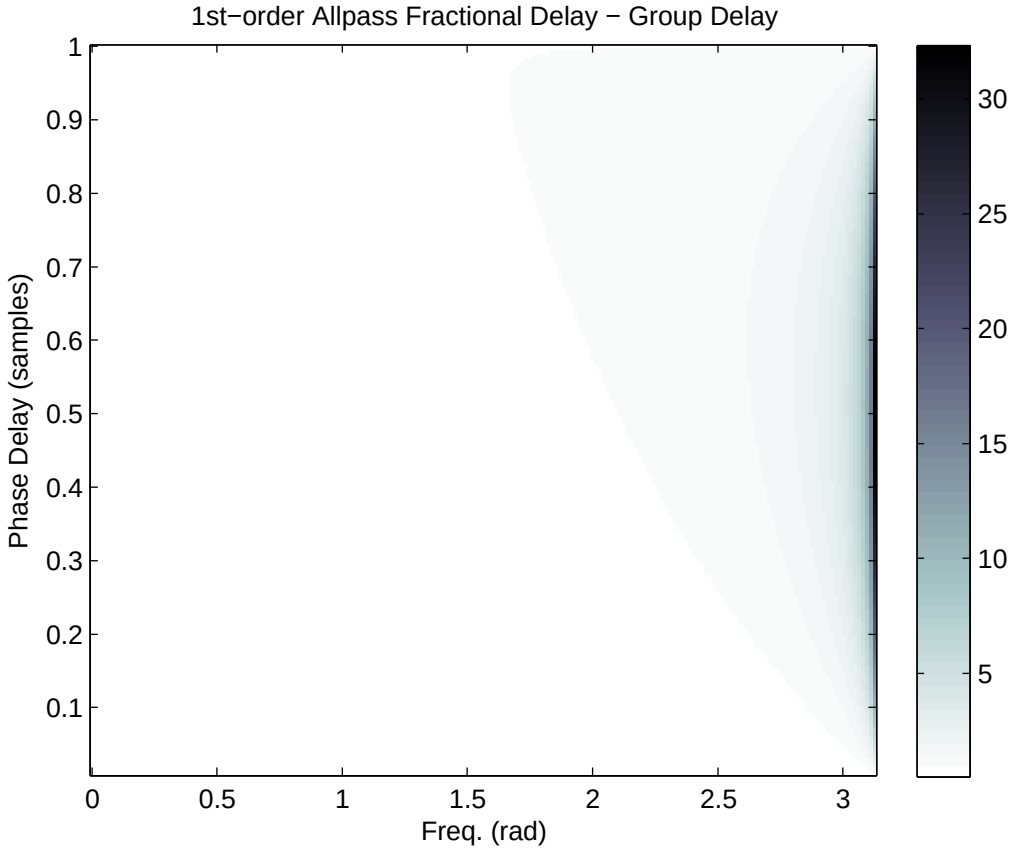


Figure B.3: Group delay at specified frequencies for which phase delay is tuned in a first-order allpass fractional delay filter.

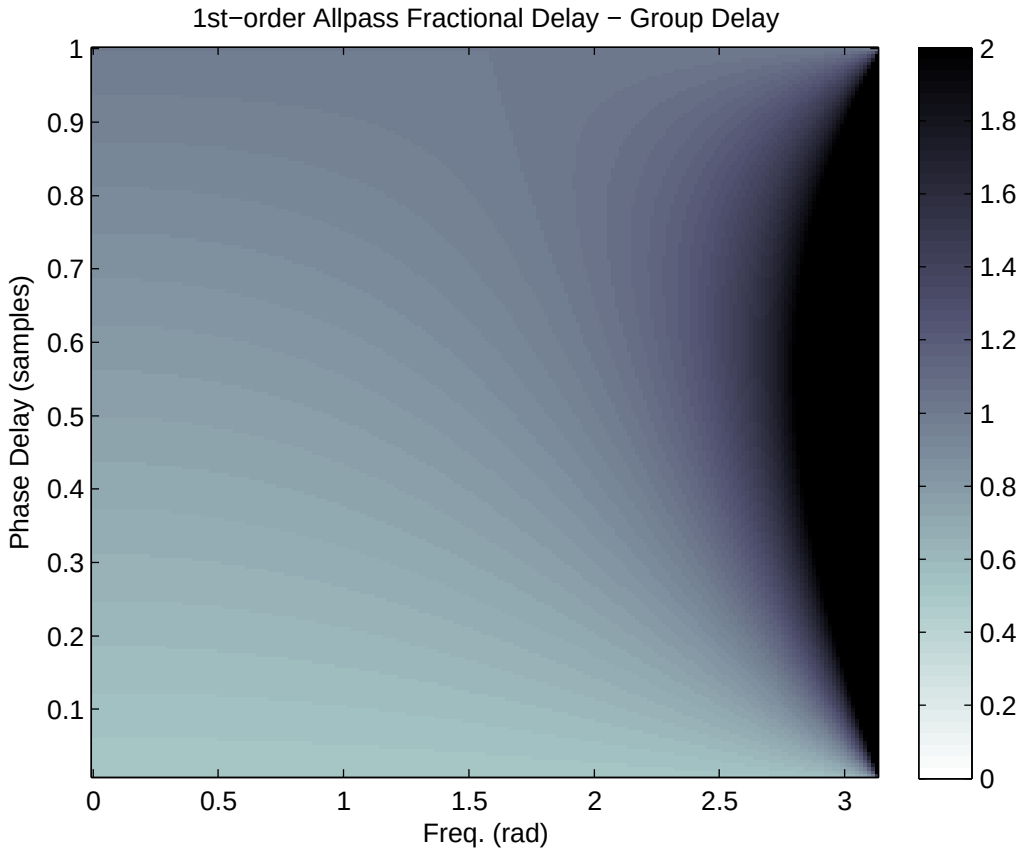


Figure B.4: Group delay at specified frequencies for which phase delay is tuned in a first-order allpass fractional delay filter. Note that the displayed values have been clipped to the range [0,2] to better display the lower range of group delays.

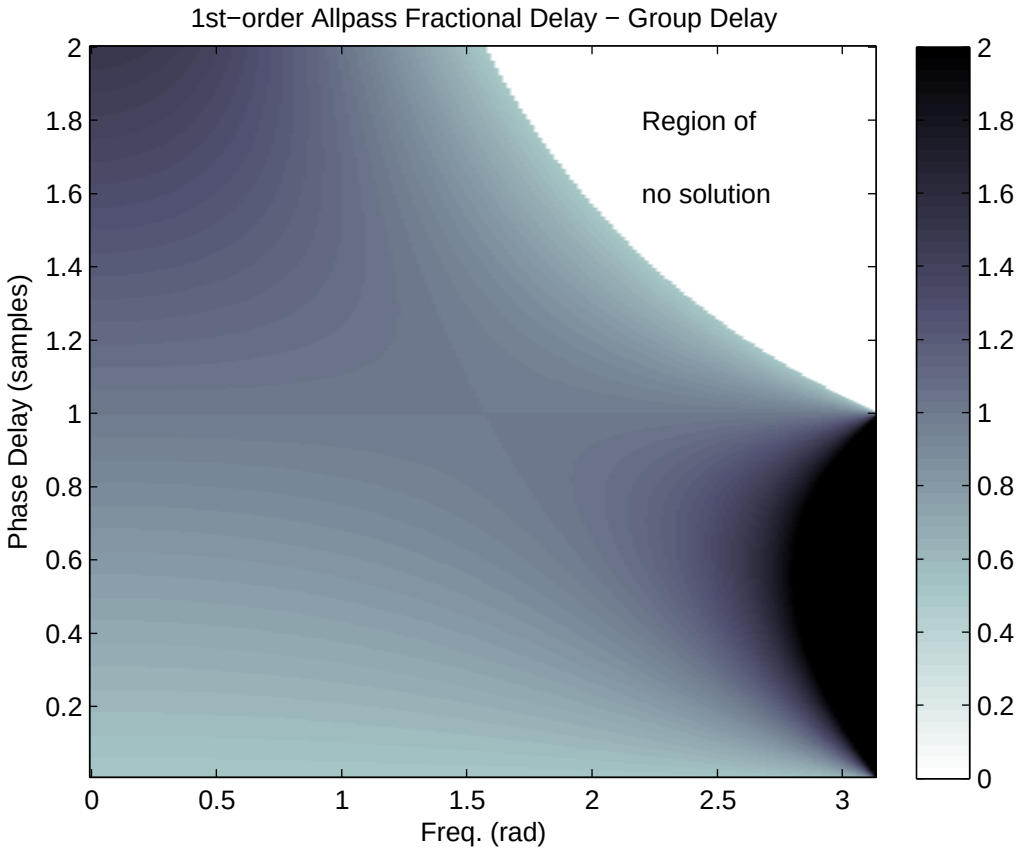


Figure B.5: Group delay at specified frequencies for which phase delay is tuned in a first-order allpass fractional delay filter.. Note that the displayed values have been clipped to the range [0,2] to better display the lower range of group delays. Also note the region for which there is no real stable filter.

APPENDIX C

BANDED WAVEGUIDE SIMULATION RESULTS

This appendix contains simulations comparing BWG and DWG models, as described in Section 4.4.5. The various BWG models are implemented as follows:

Type A The BPFs are designed using just the string loop.

- **Type A1** Phase delay is set to correspond to a multiple of each modal frequency's wavelength.
- **Type A2** Phase delay at each modal frequency is equal to that of the DWG.
- **Type A3** Parameters, including modal frequency, are set to correspond to an ideal harmonic model.

Type B The BPFs are designed using the string loop and excitation position filter.

- **Type B1** Phase delay is set to correspond to a multiple of each modal frequency's wavelength.
- **Type B2** Phase delay at each modal frequency is equal to that of the DWG.
- **Type B3** Parameters, including modal frequency, are set to correspond to an ideal harmonic model.

C.1 Bowed String Tuned to 441 Hz with no Stiffness

The following figures show the velocity amplitude envelope, steady-state oscillations, and spectrograms of various, comparably designed, BWG and DWG models. All models are of a string tuned to 441 Hz with no stiffness, as described in Section 4.4.5.

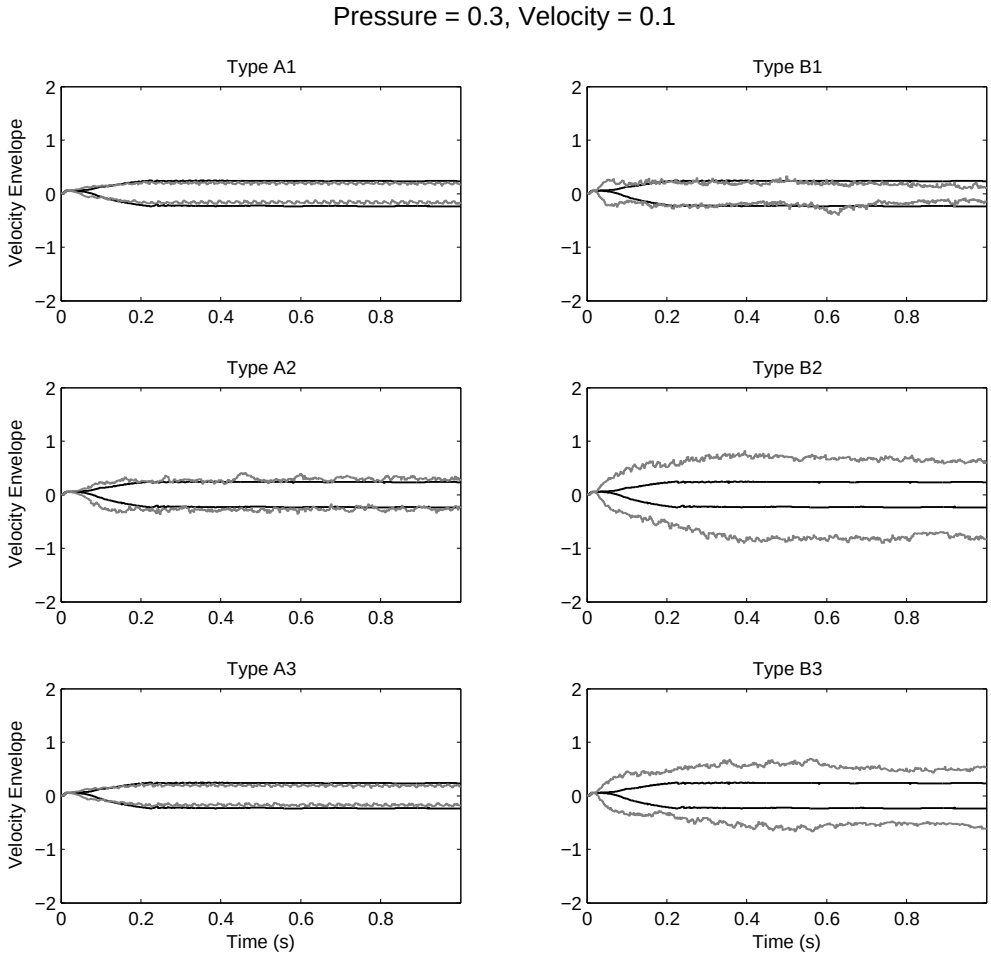


Figure C.1: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

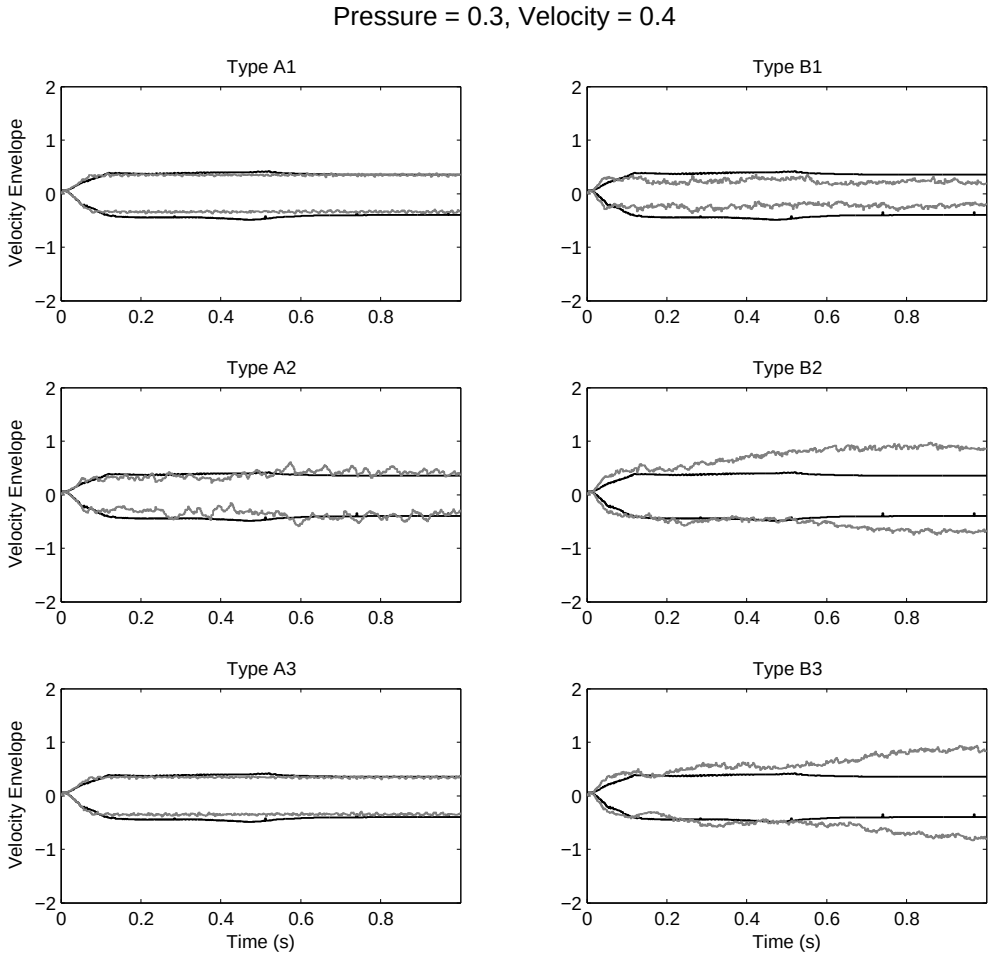


Figure C.2: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

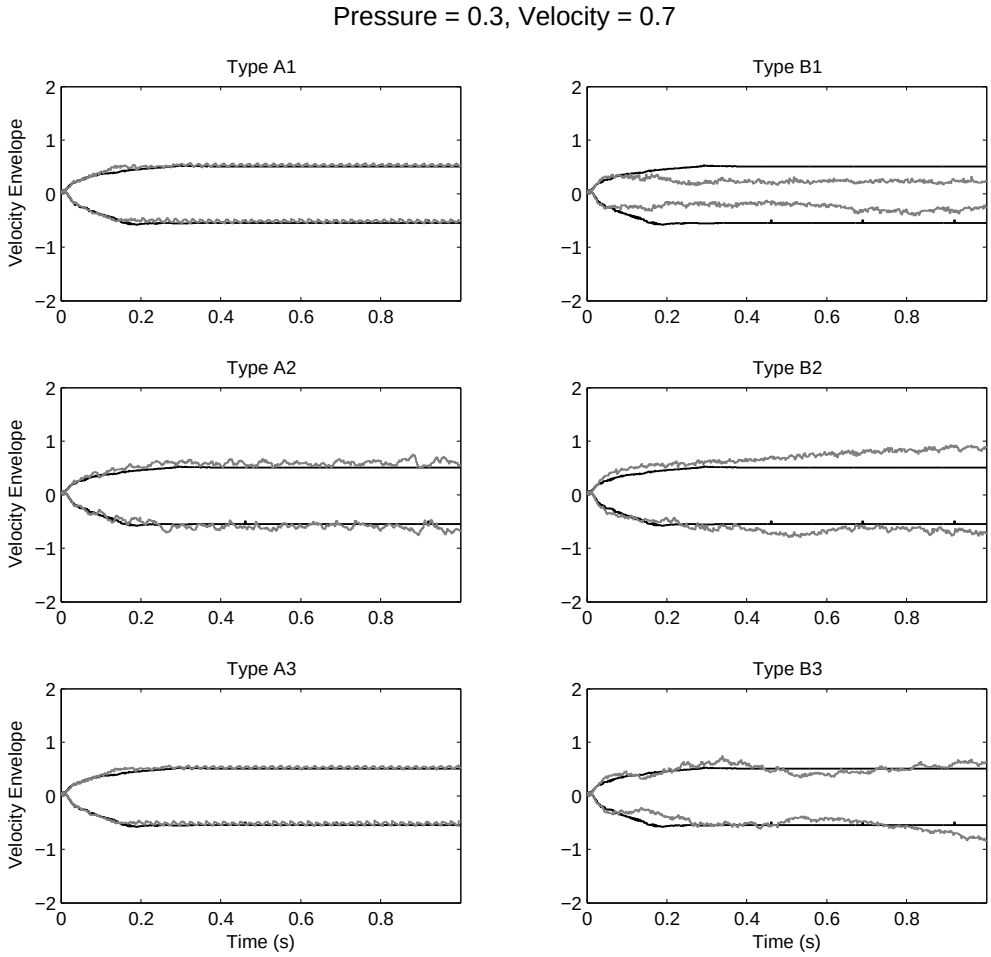


Figure C.3: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

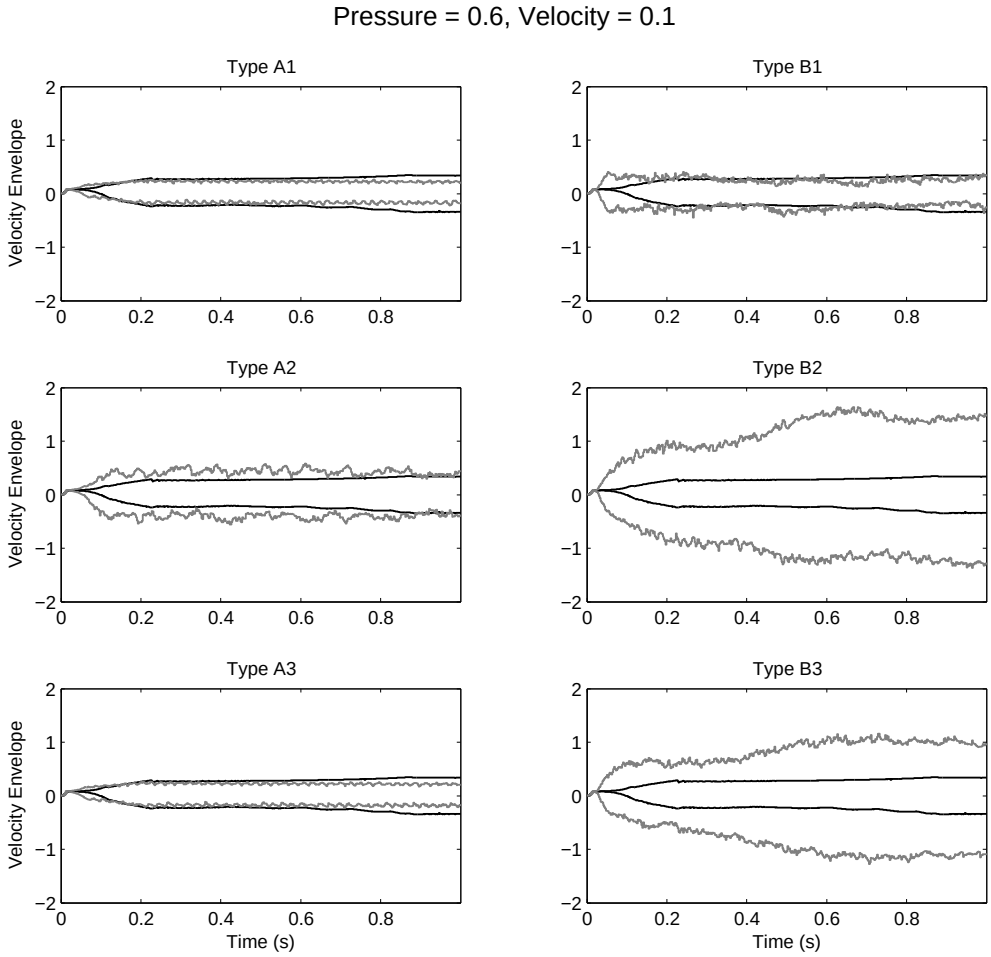


Figure C.4: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

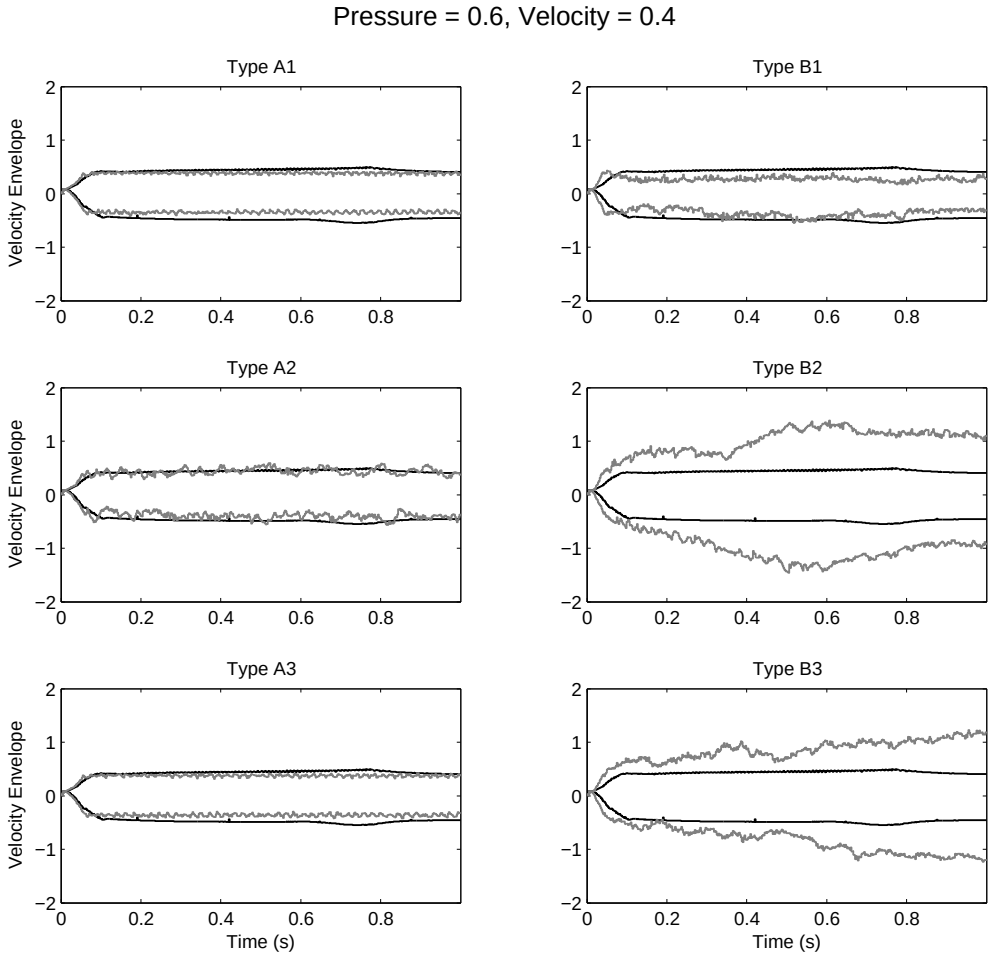


Figure C.5: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

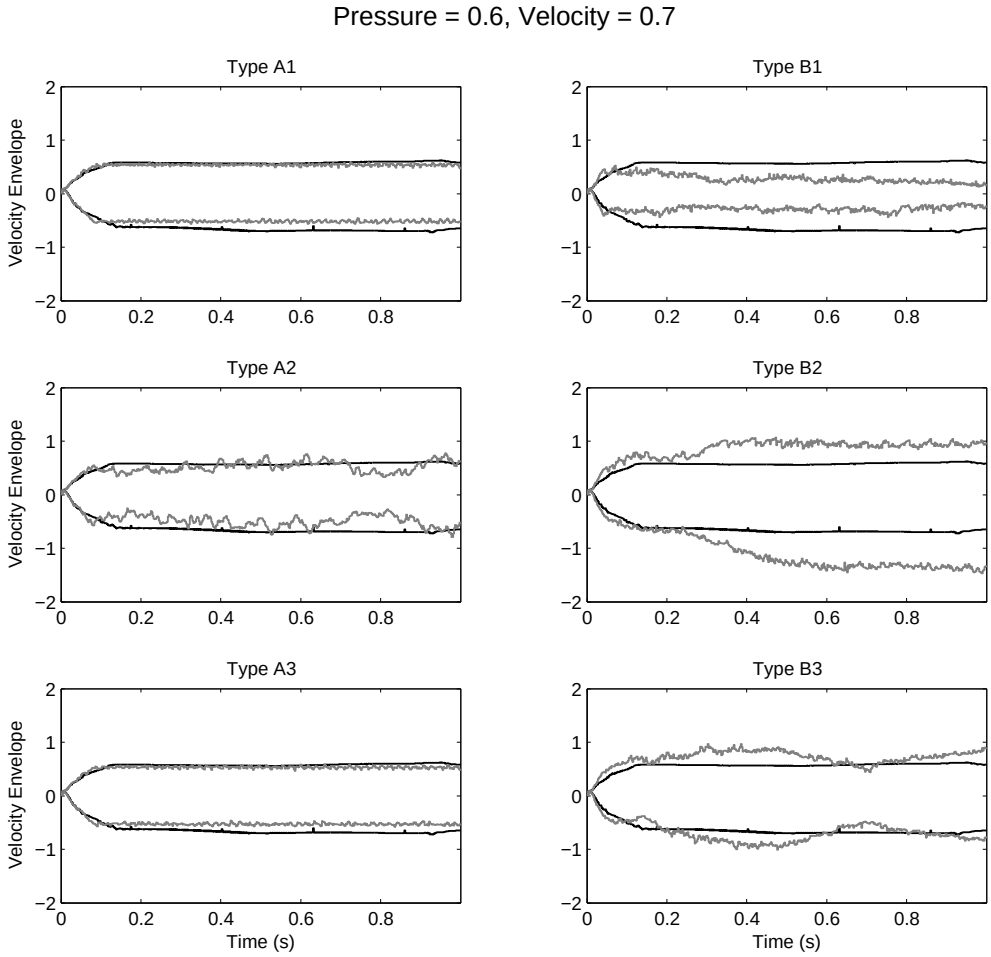


Figure C.6: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

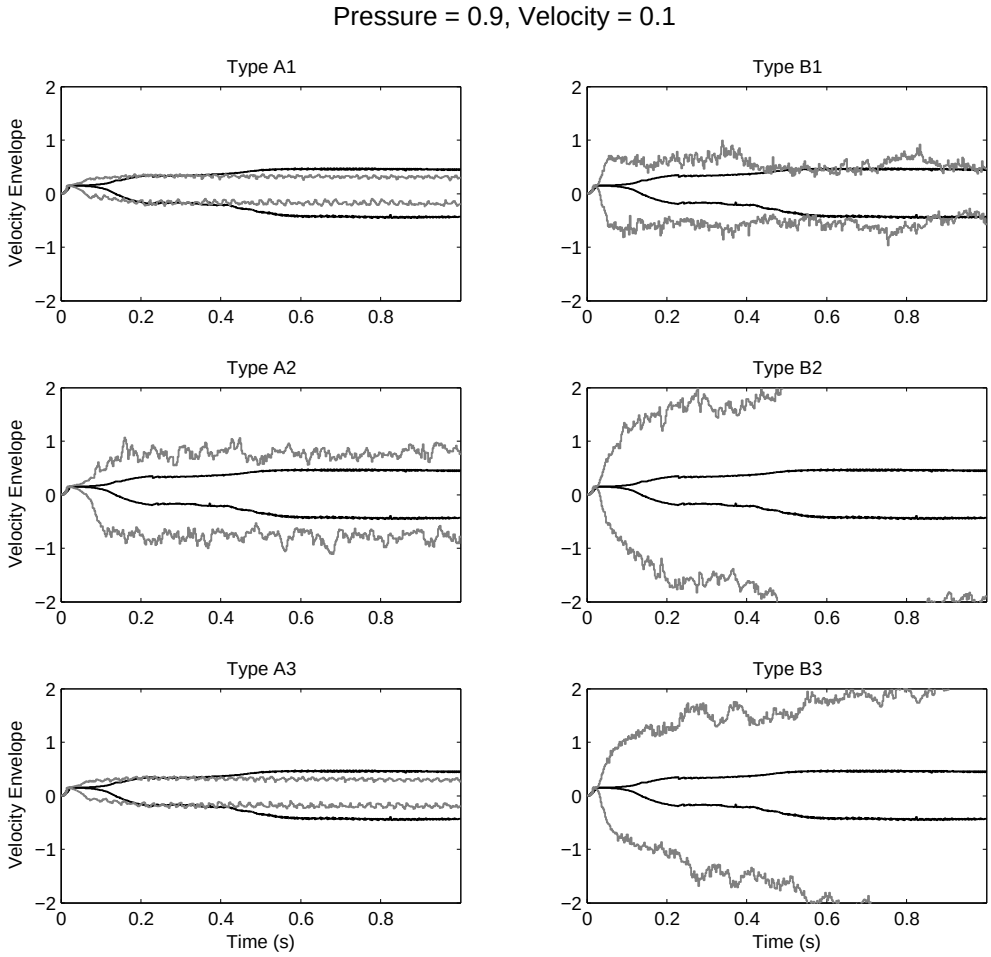


Figure C.7: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

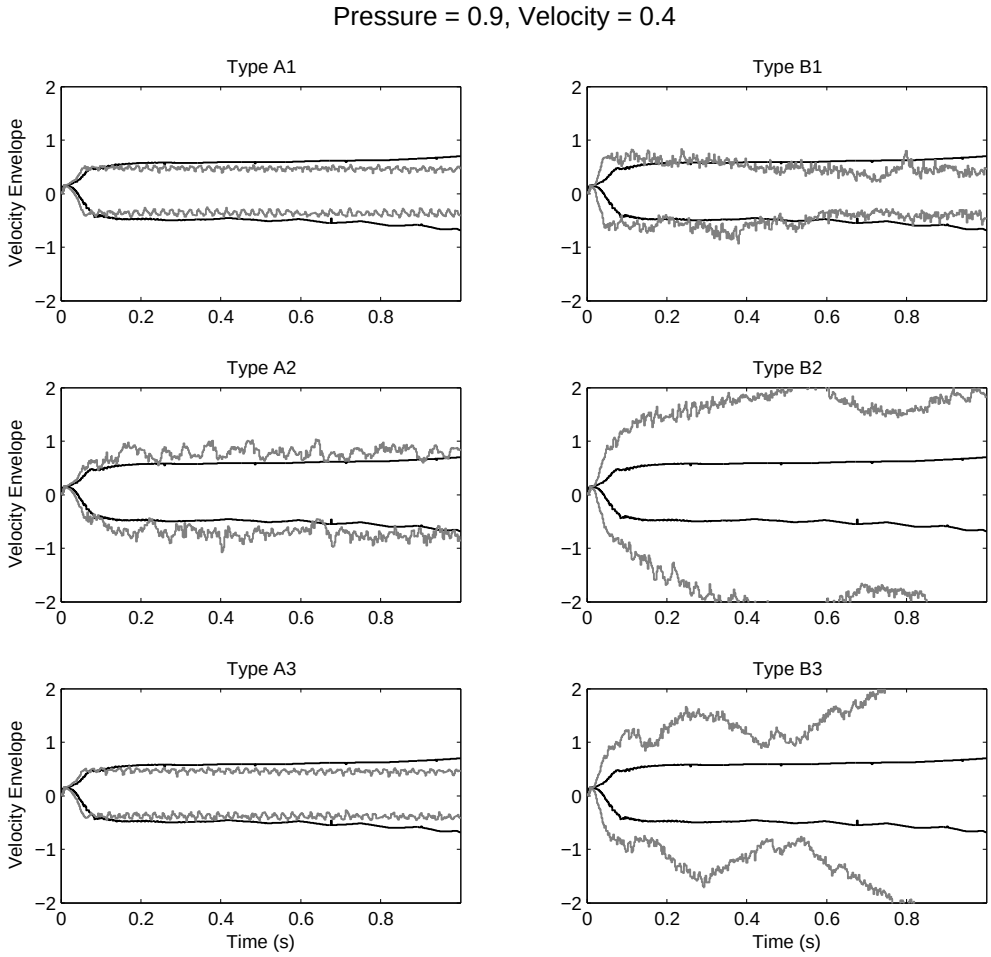


Figure C.8: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

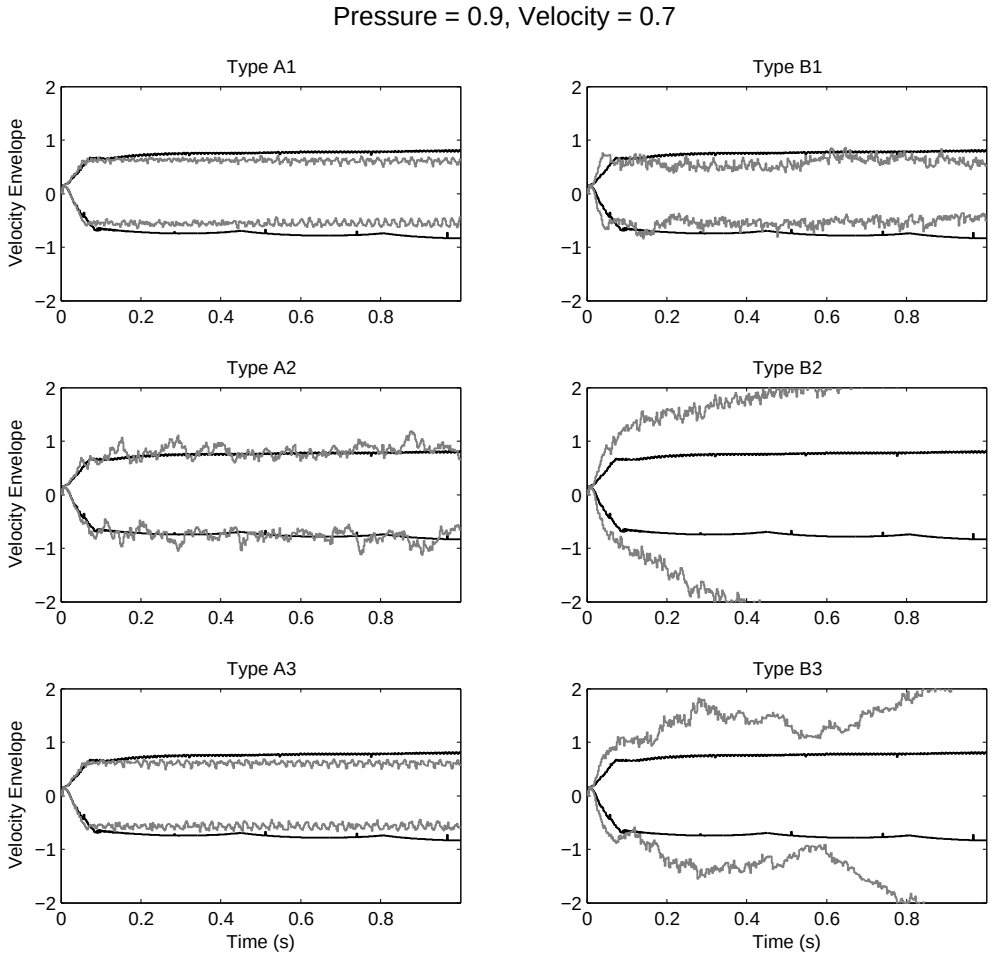


Figure C.9: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

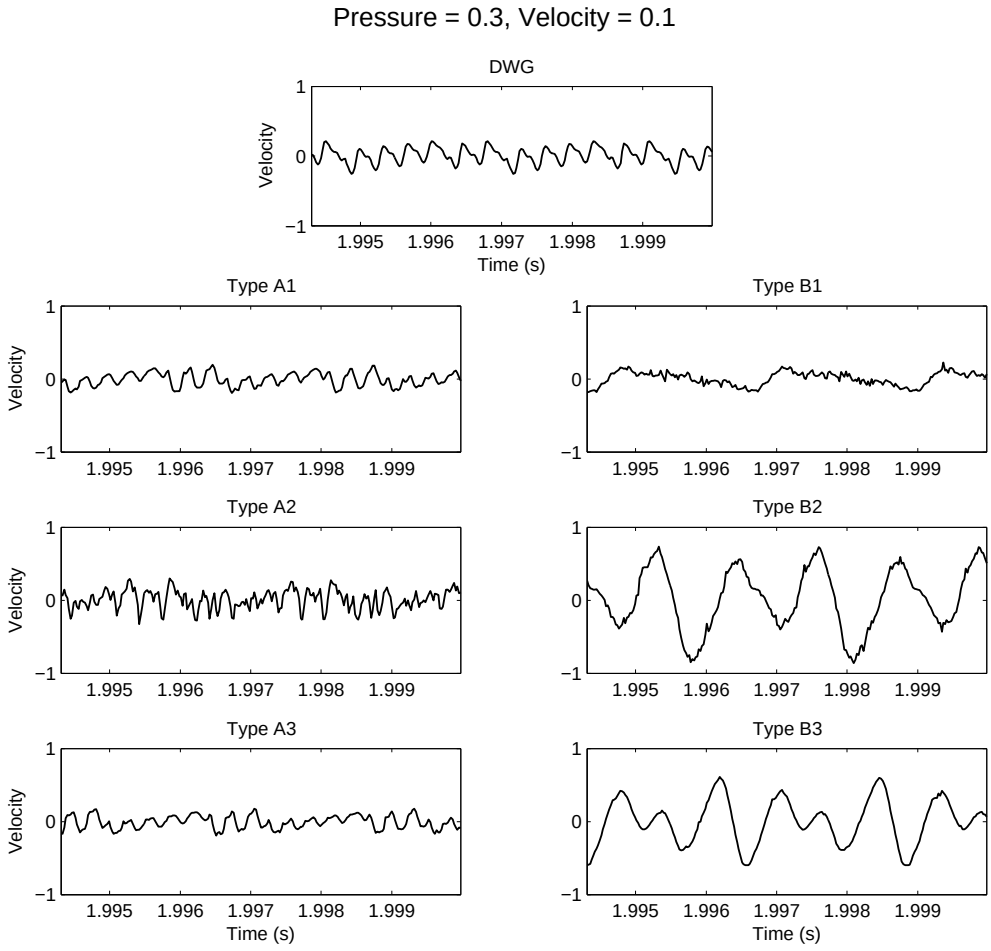


Figure C.10: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

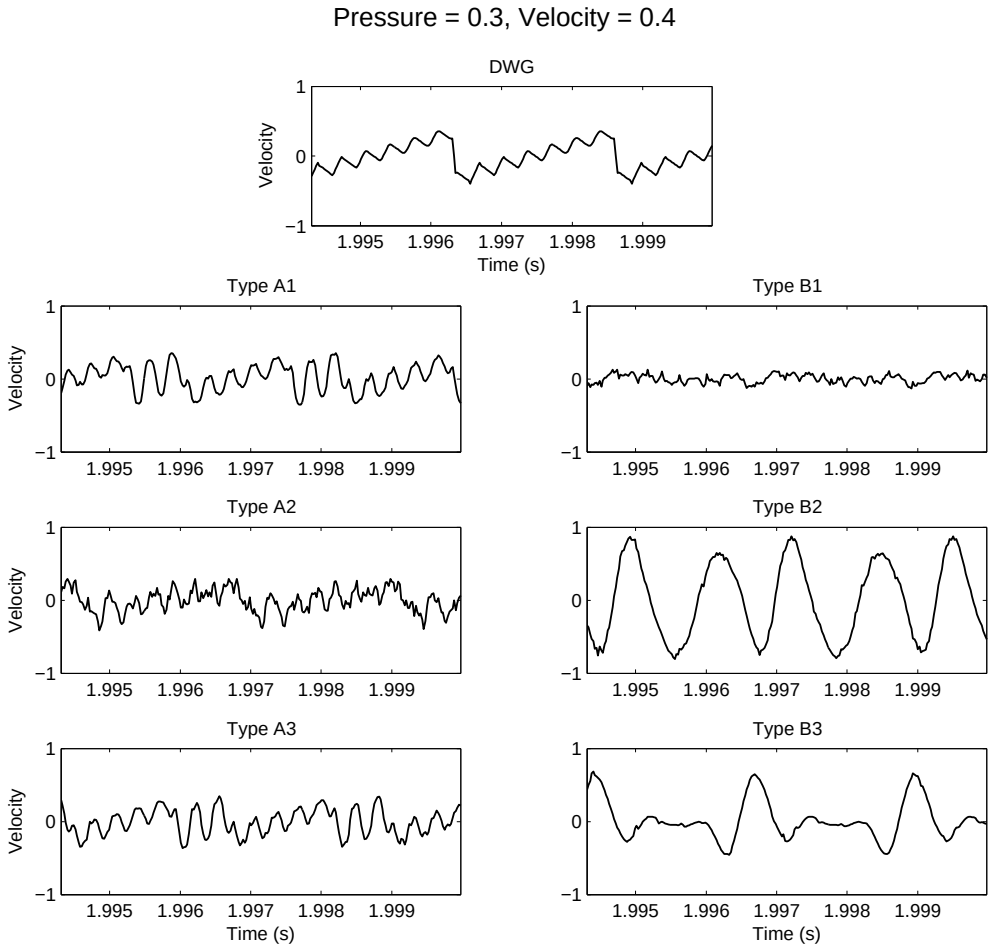


Figure C.11: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

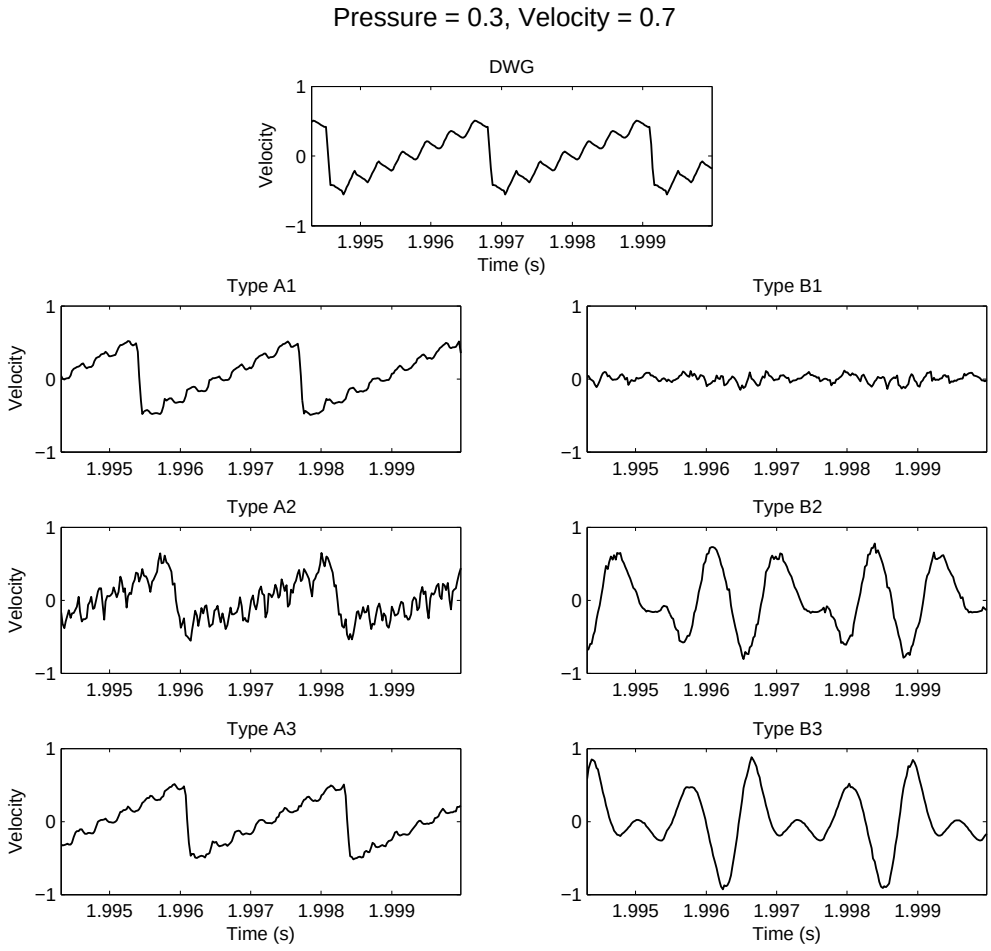


Figure C.12: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

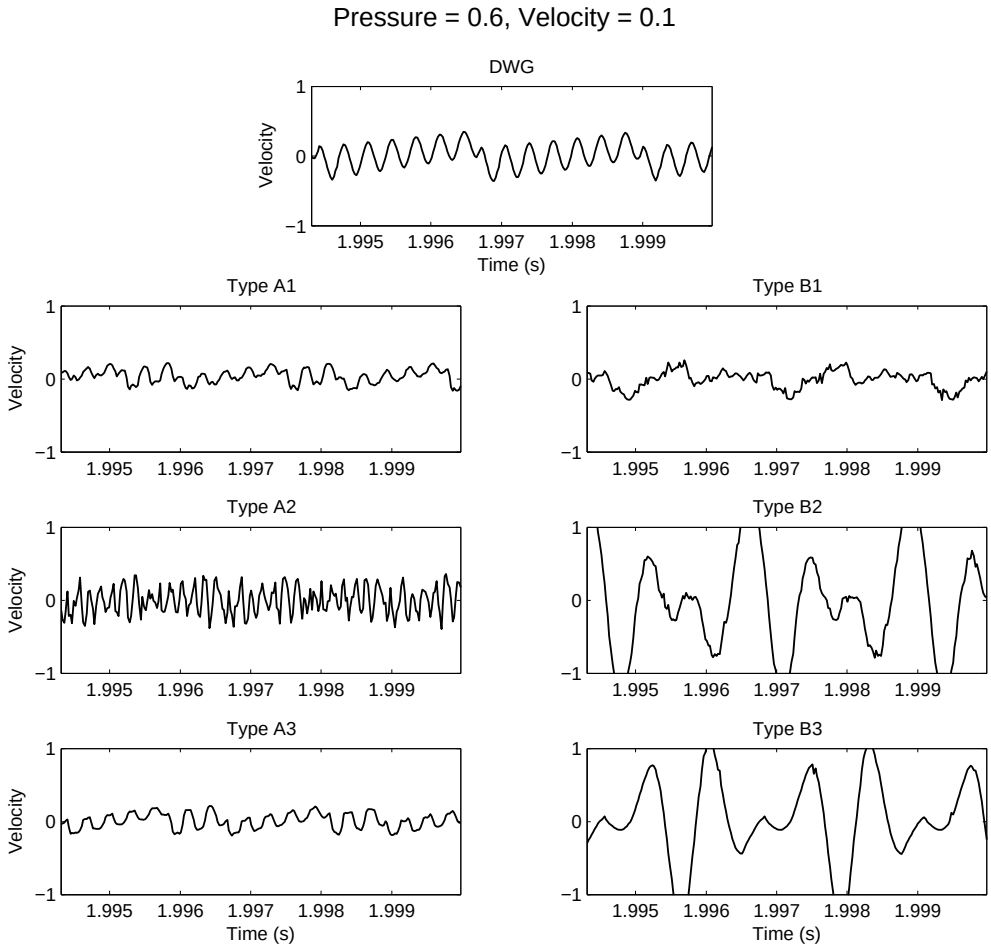


Figure C.13: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

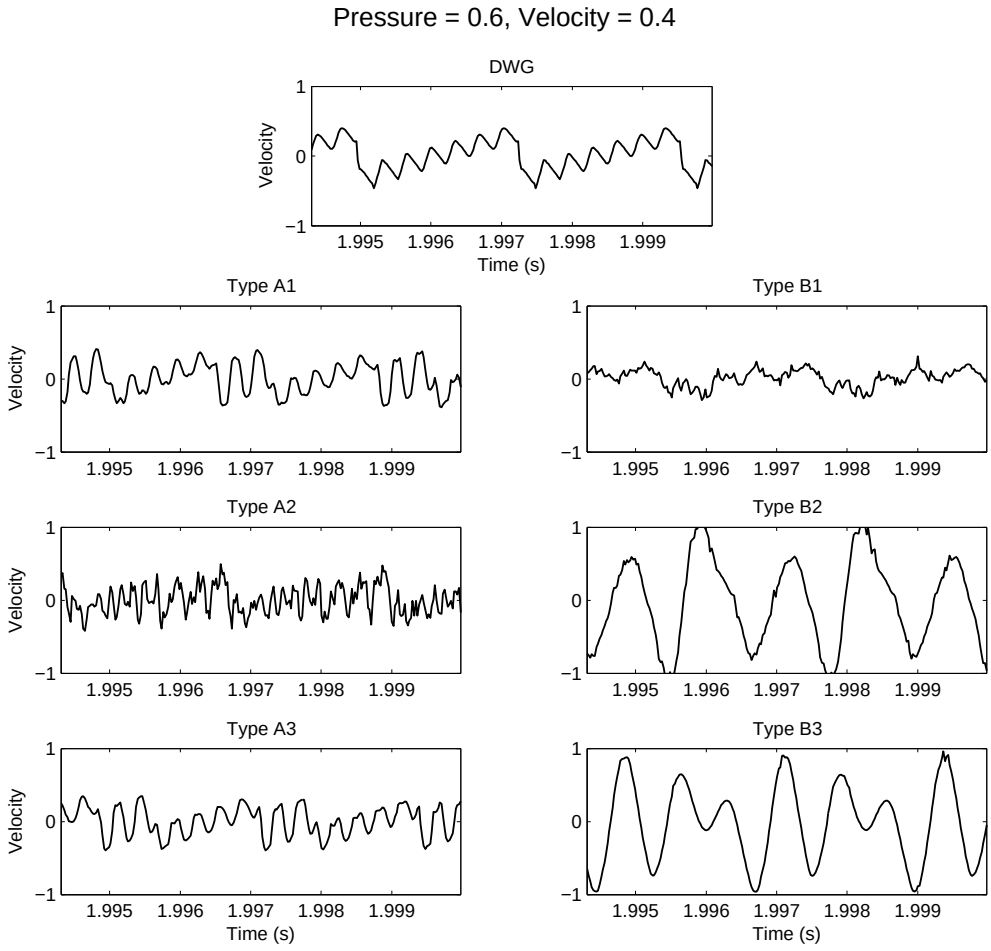


Figure C.14: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

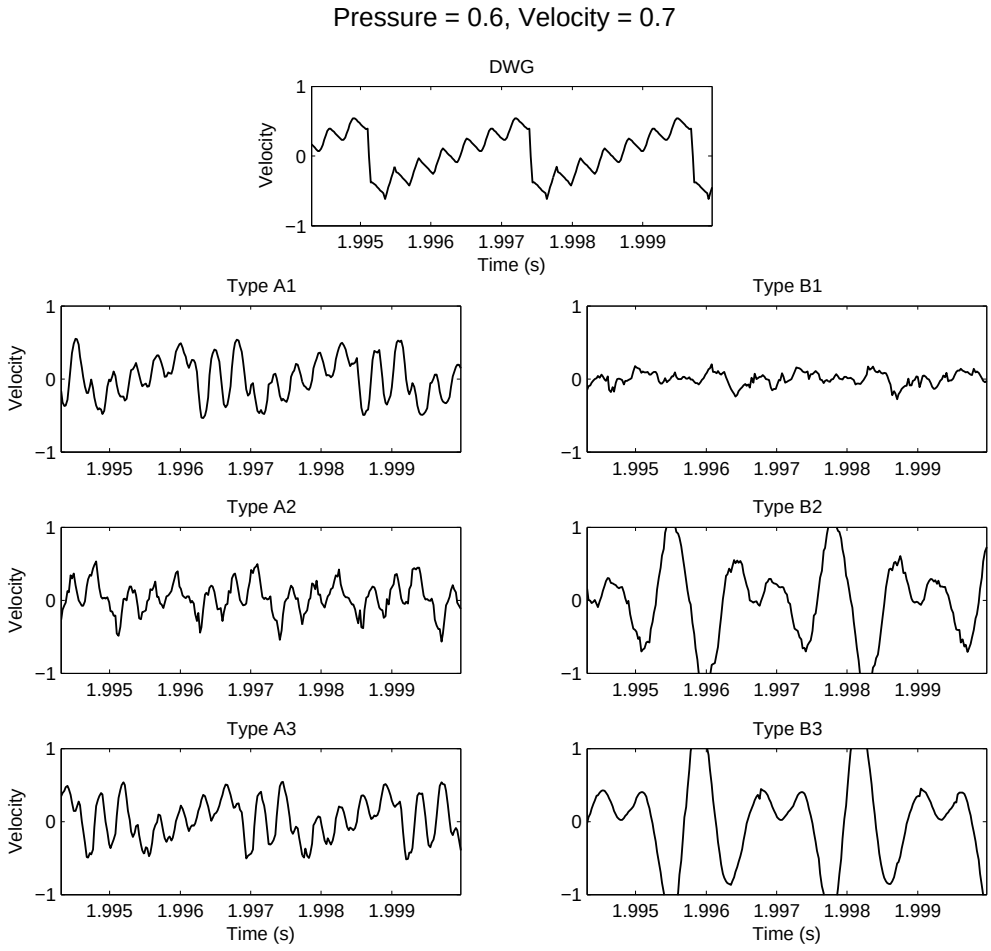


Figure C.15: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

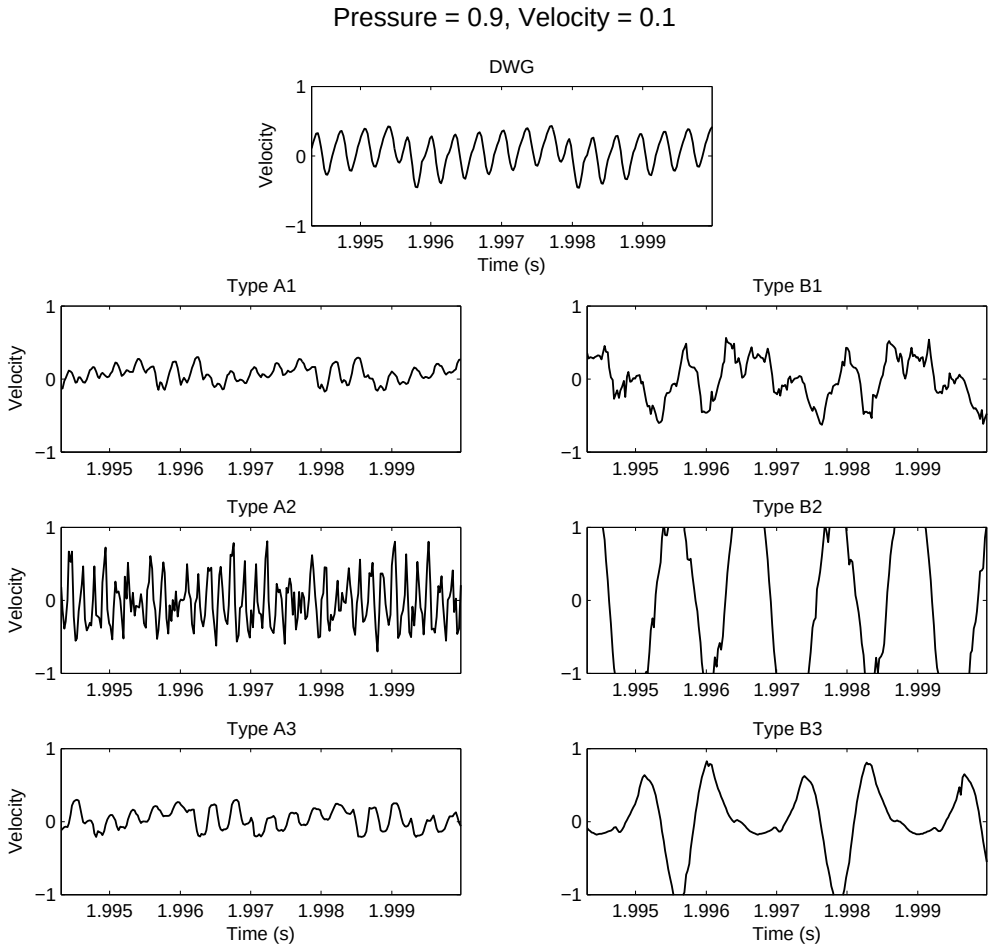


Figure C.16: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

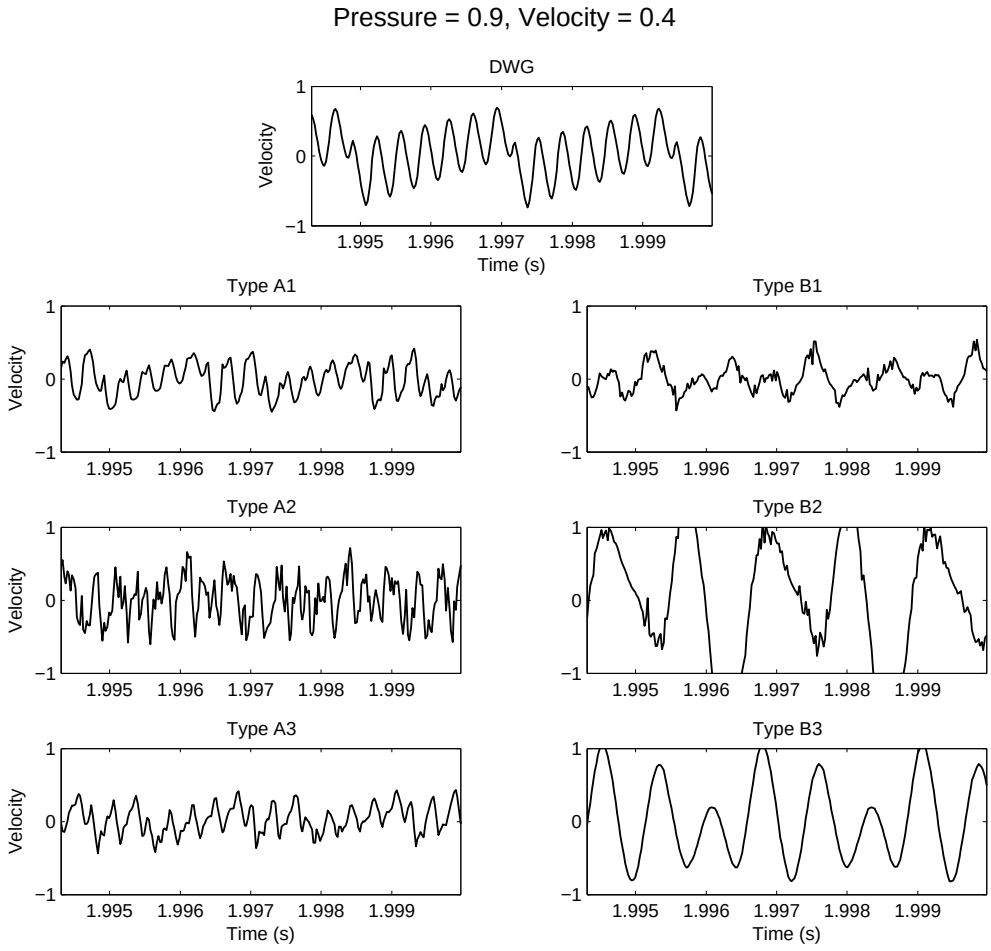


Figure C.17: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

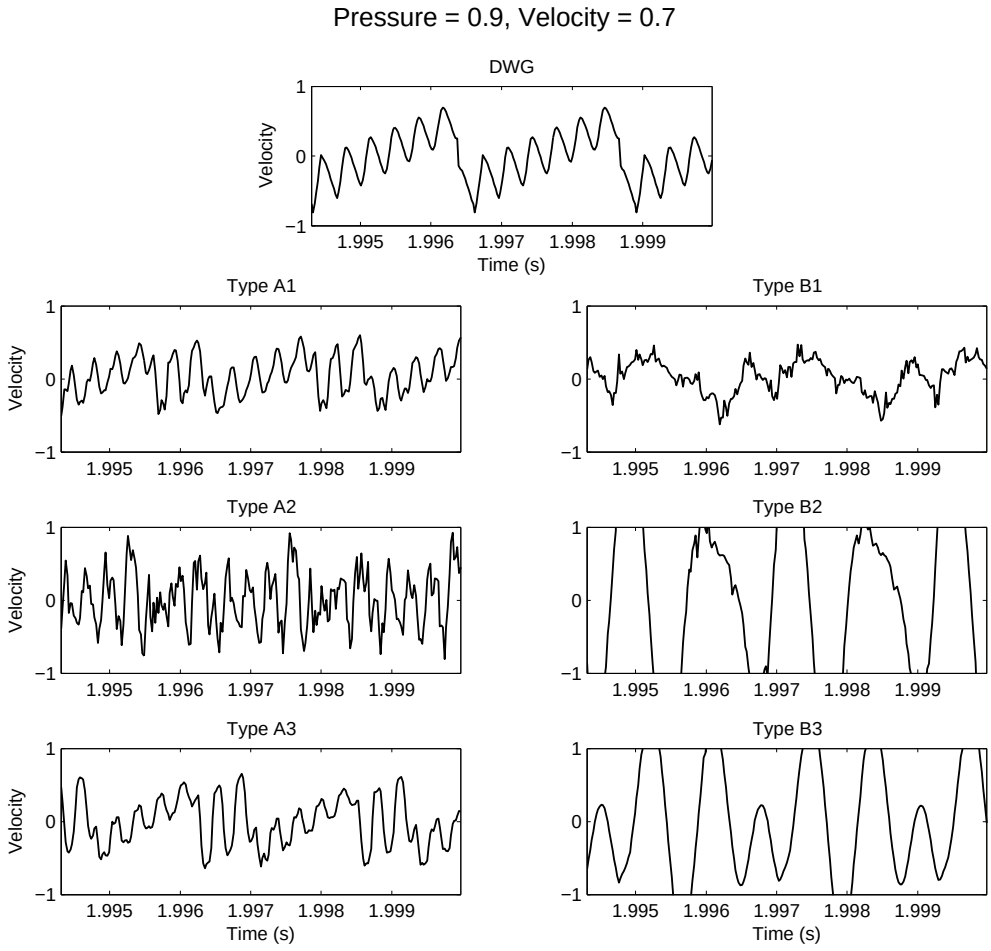


Figure C.18: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

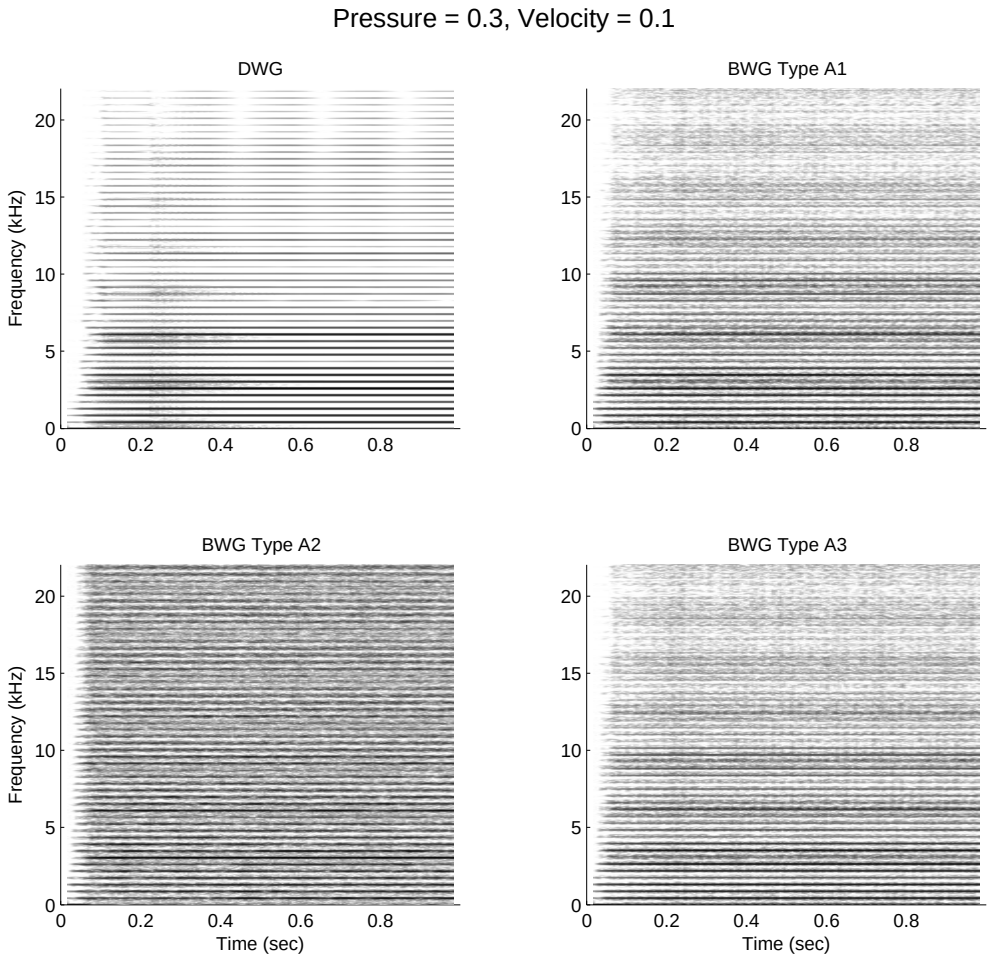


Figure C.19: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

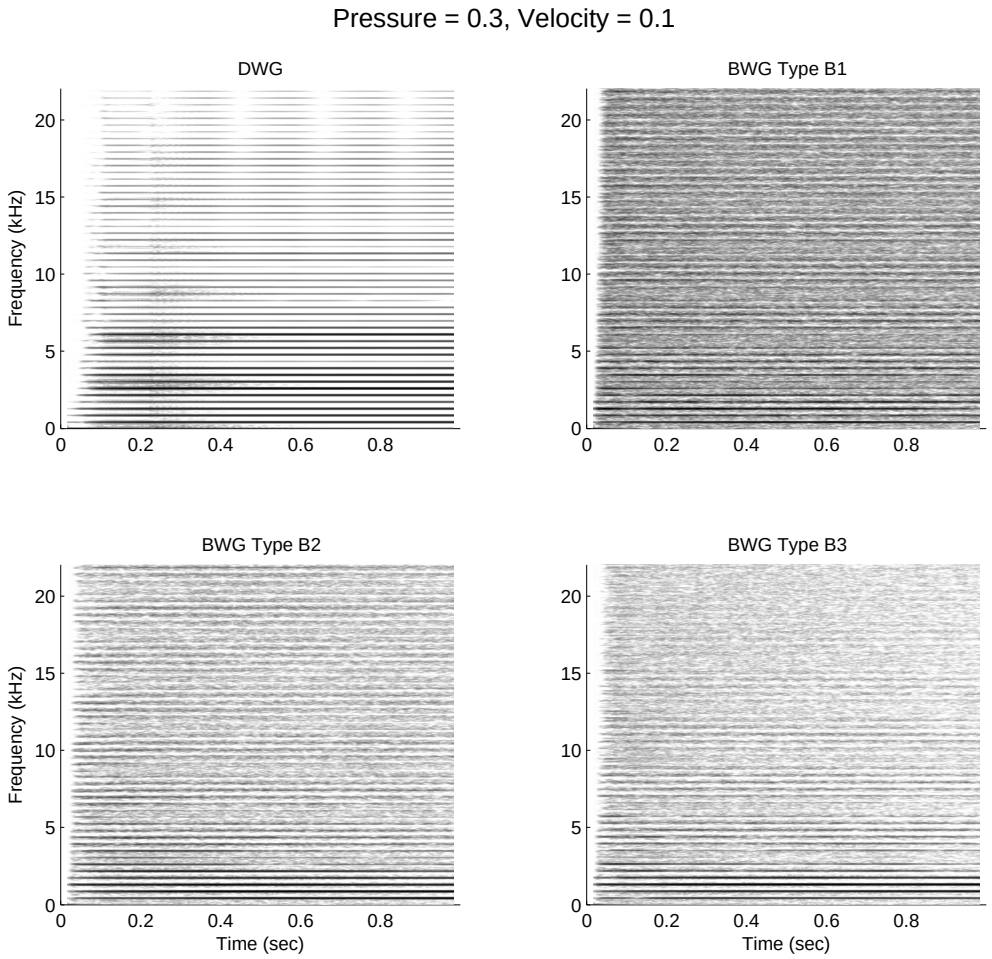


Figure C.20: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

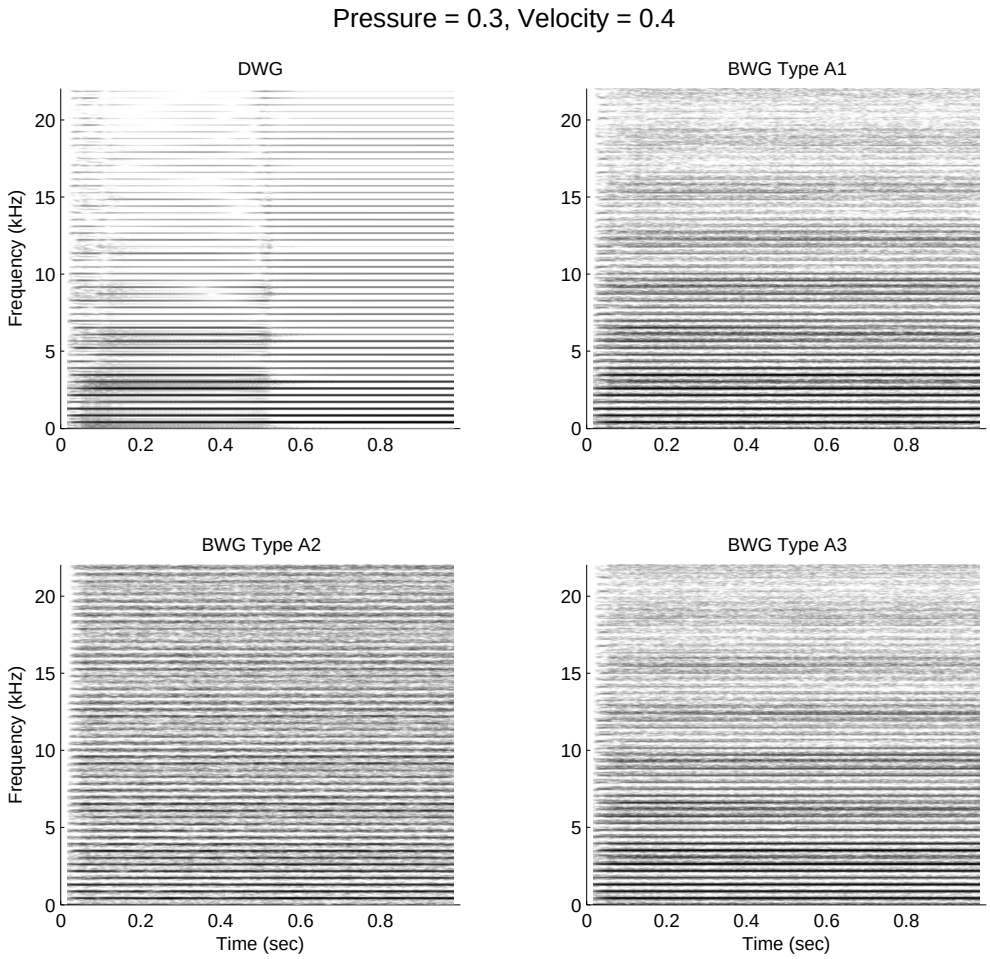


Figure C.21: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

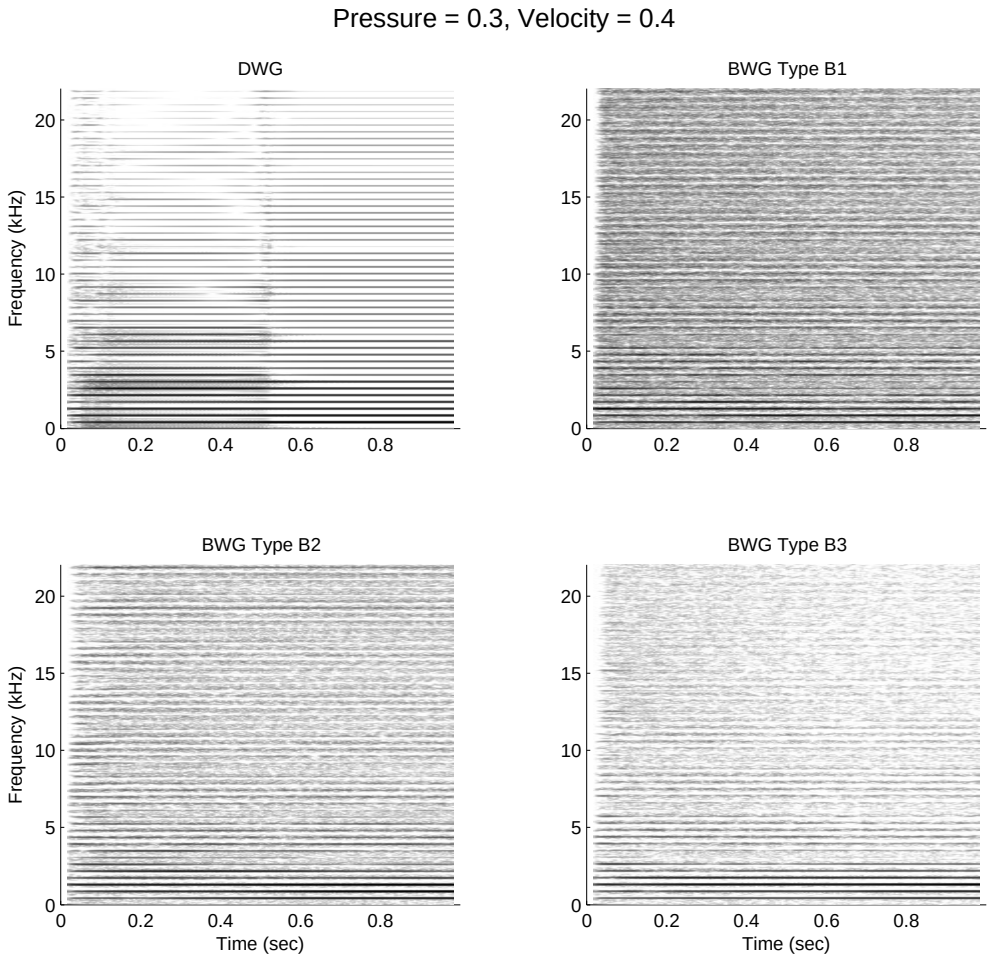


Figure C.22: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

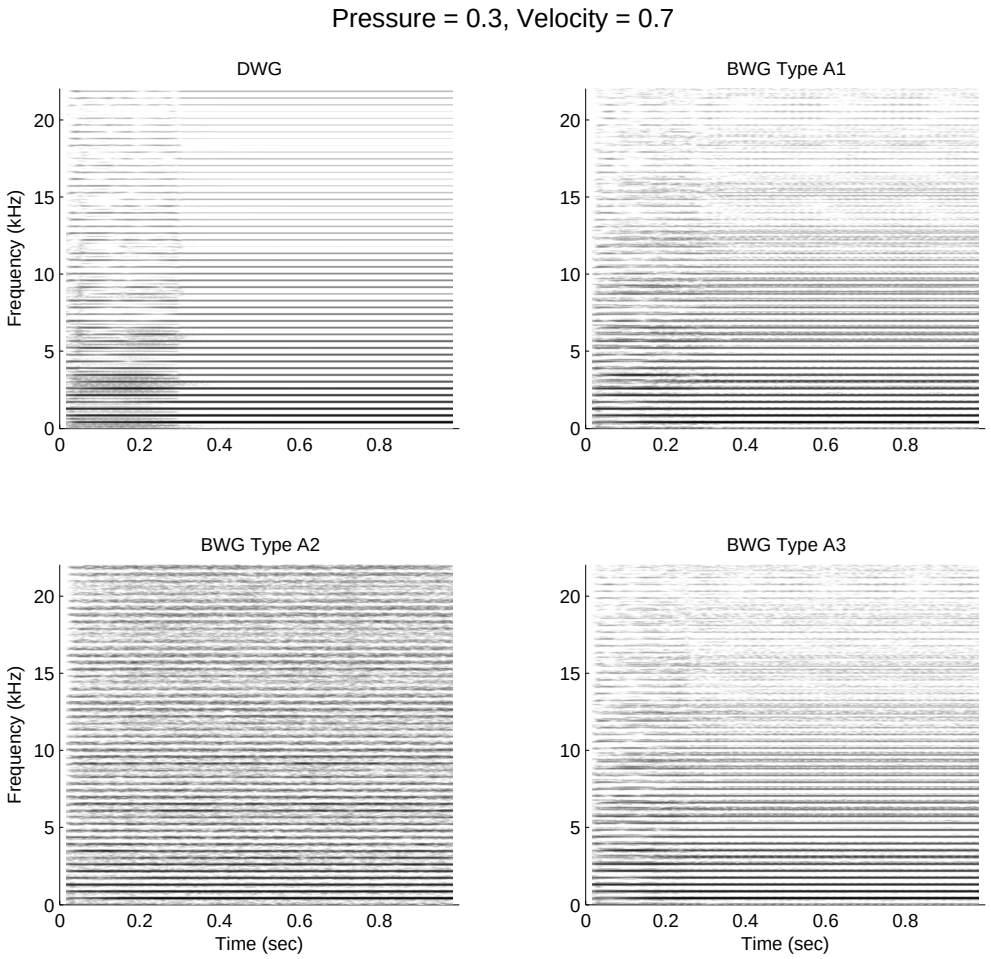


Figure C.23: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

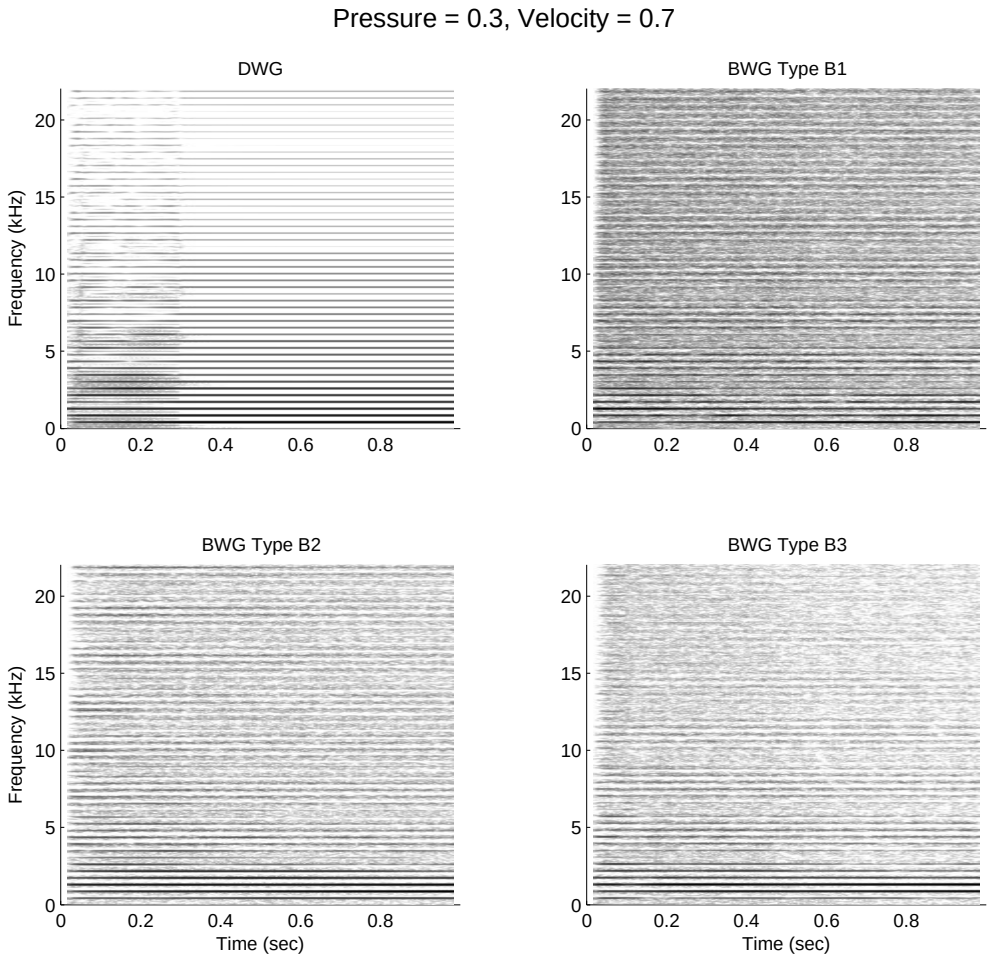


Figure C.24: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

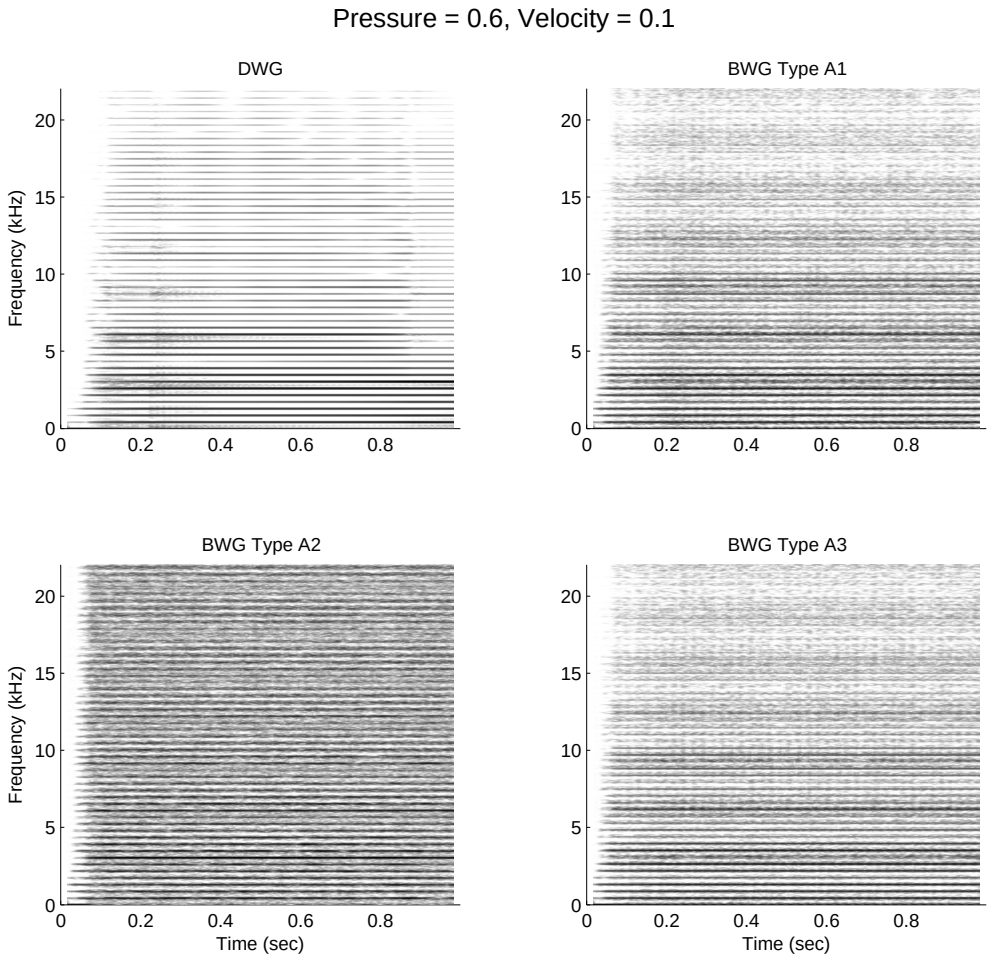


Figure C.25: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

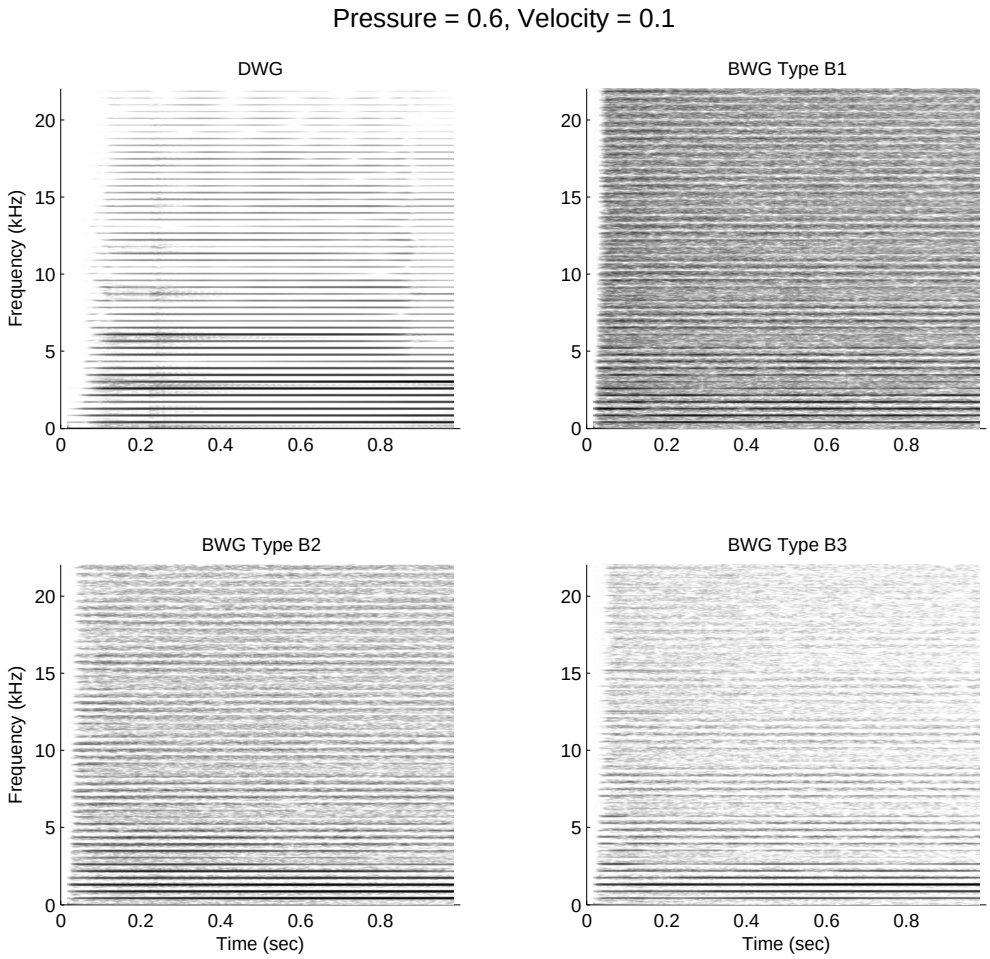


Figure C.26: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

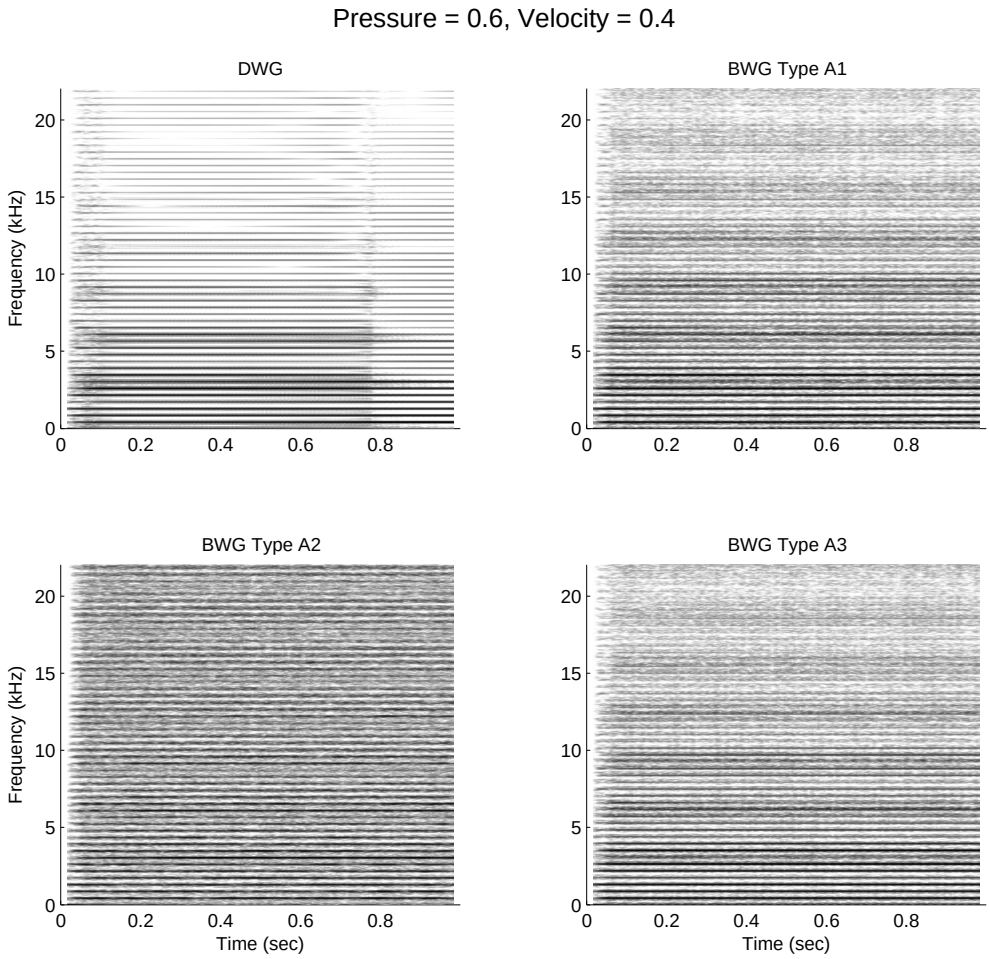


Figure C.27: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

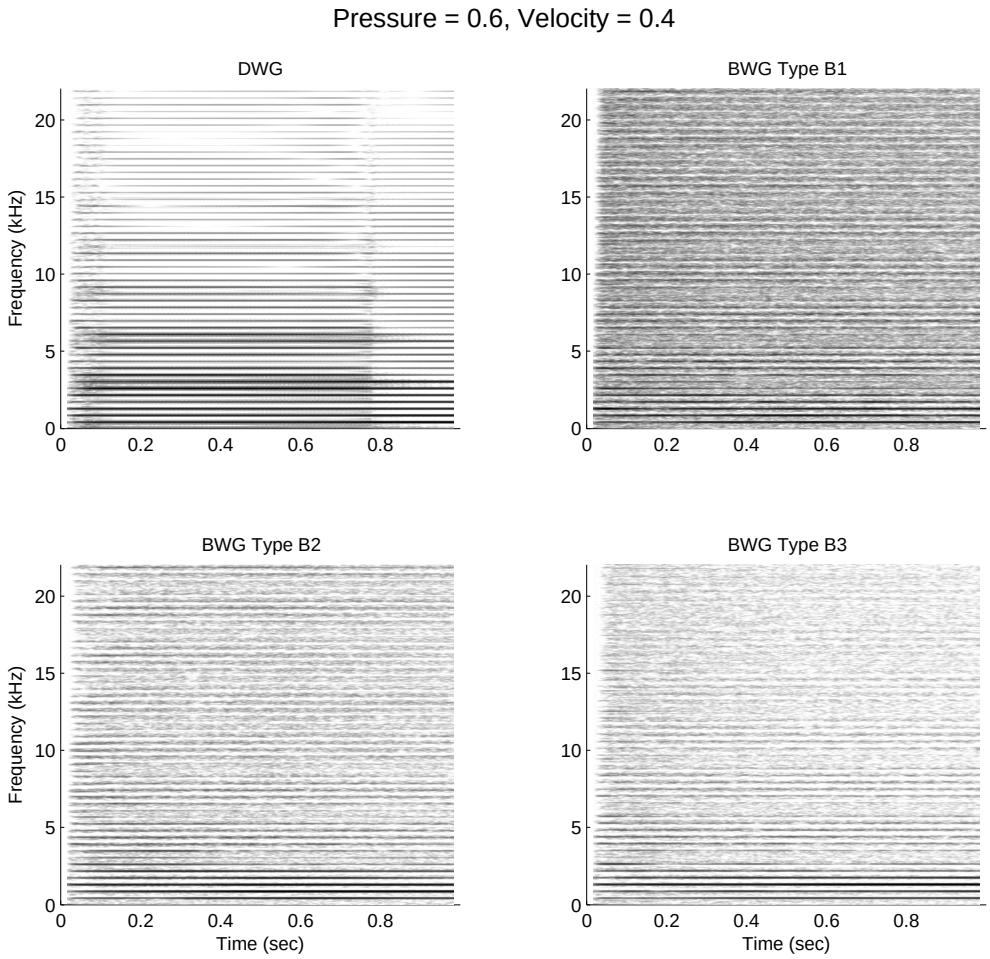


Figure C.28: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

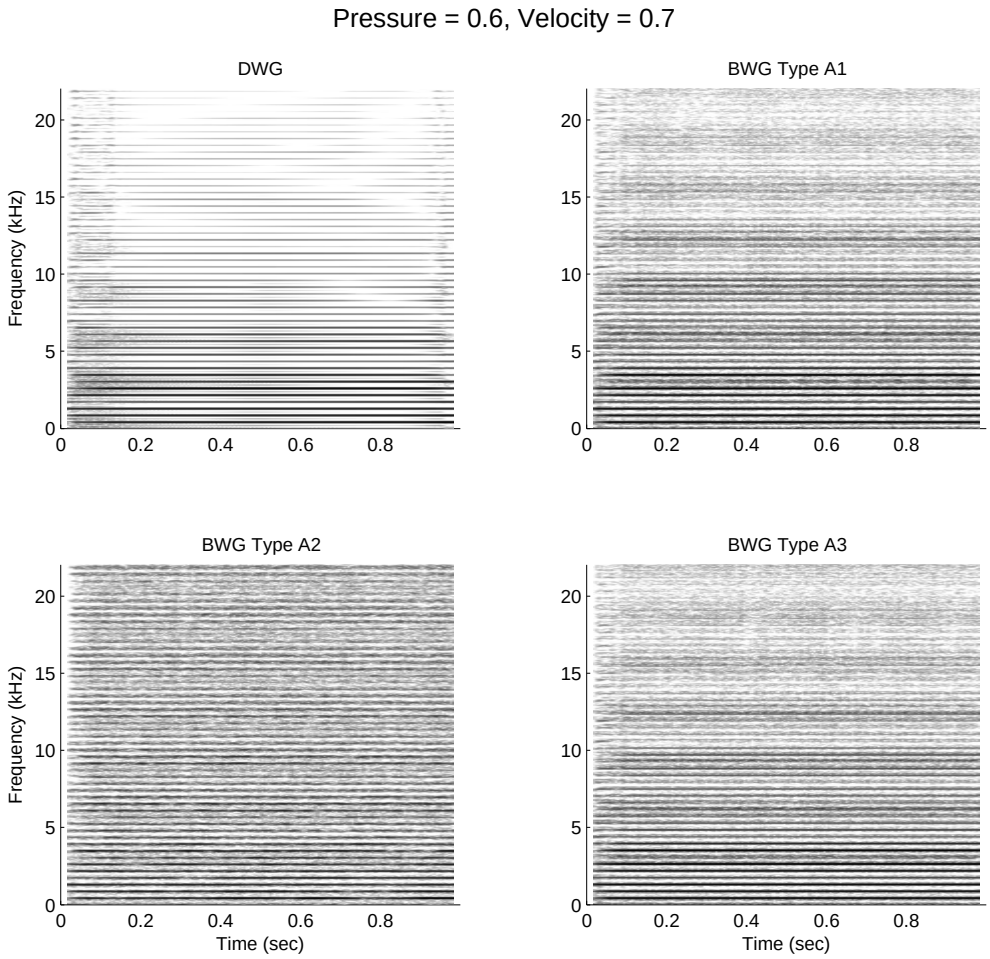


Figure C.29: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

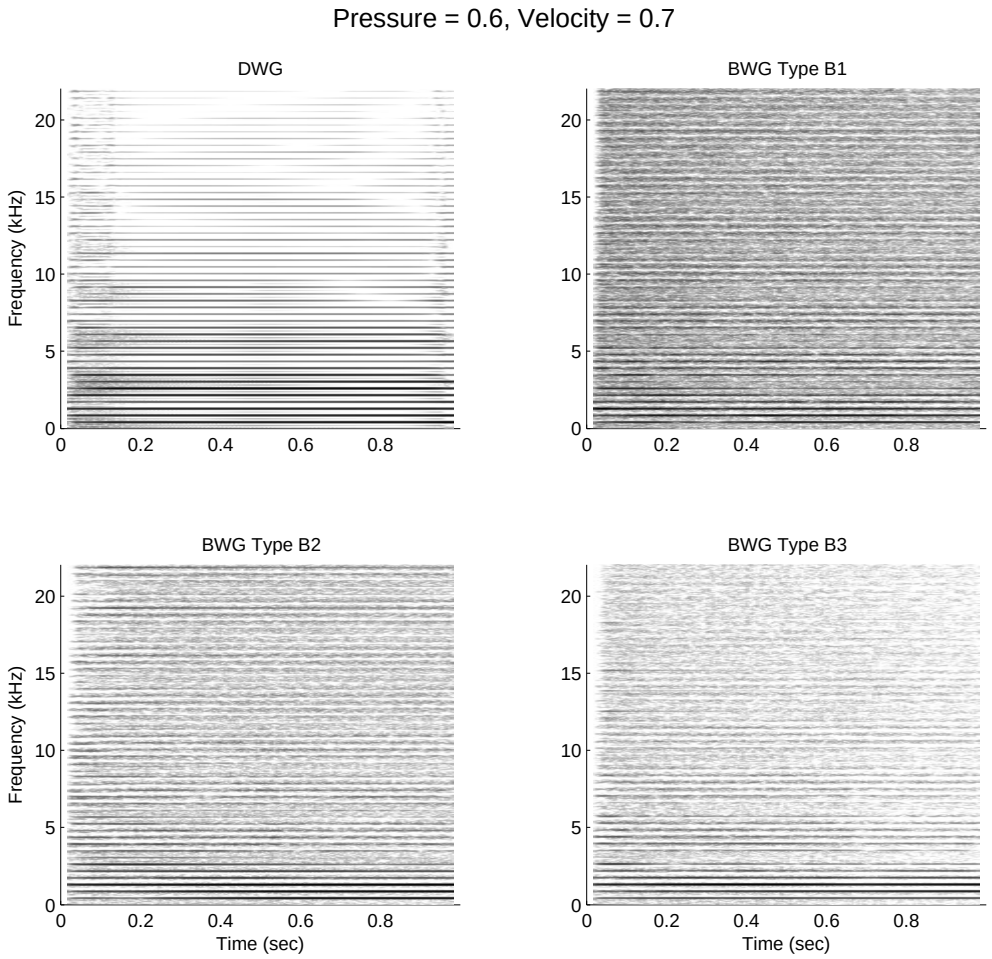


Figure C.30: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

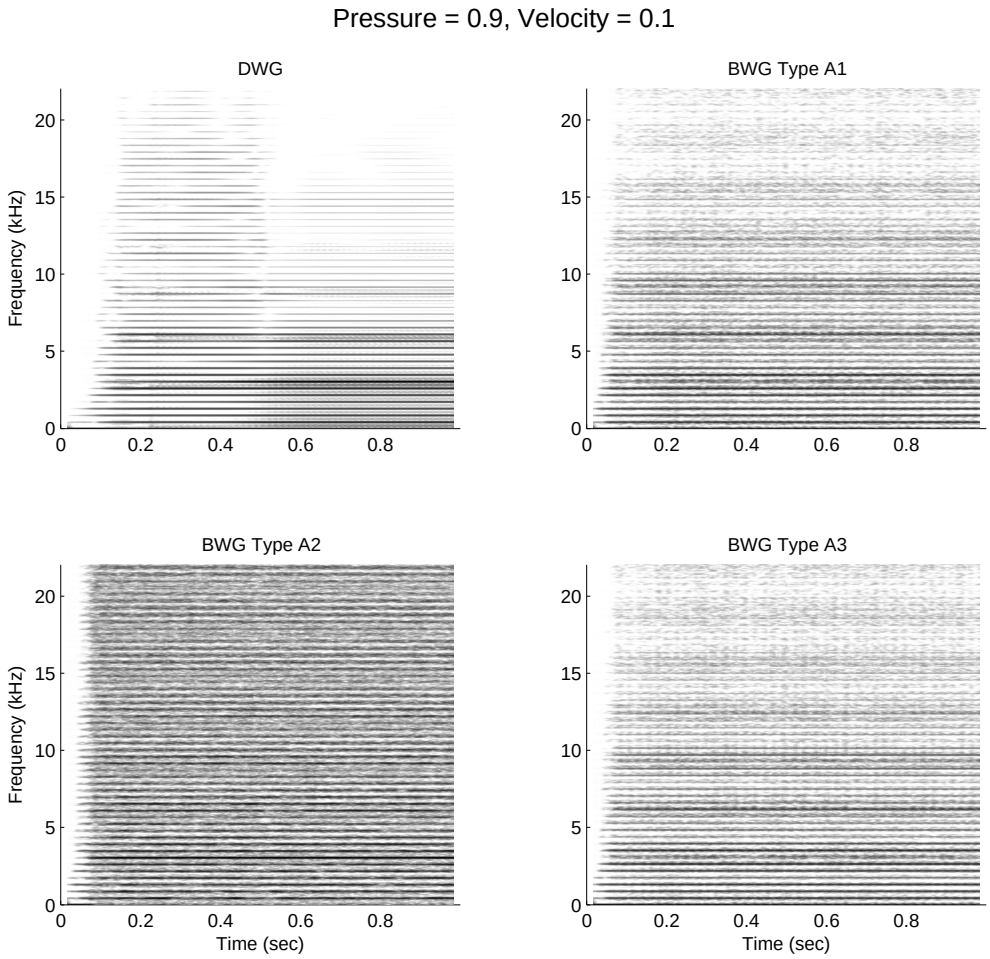


Figure C.31: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

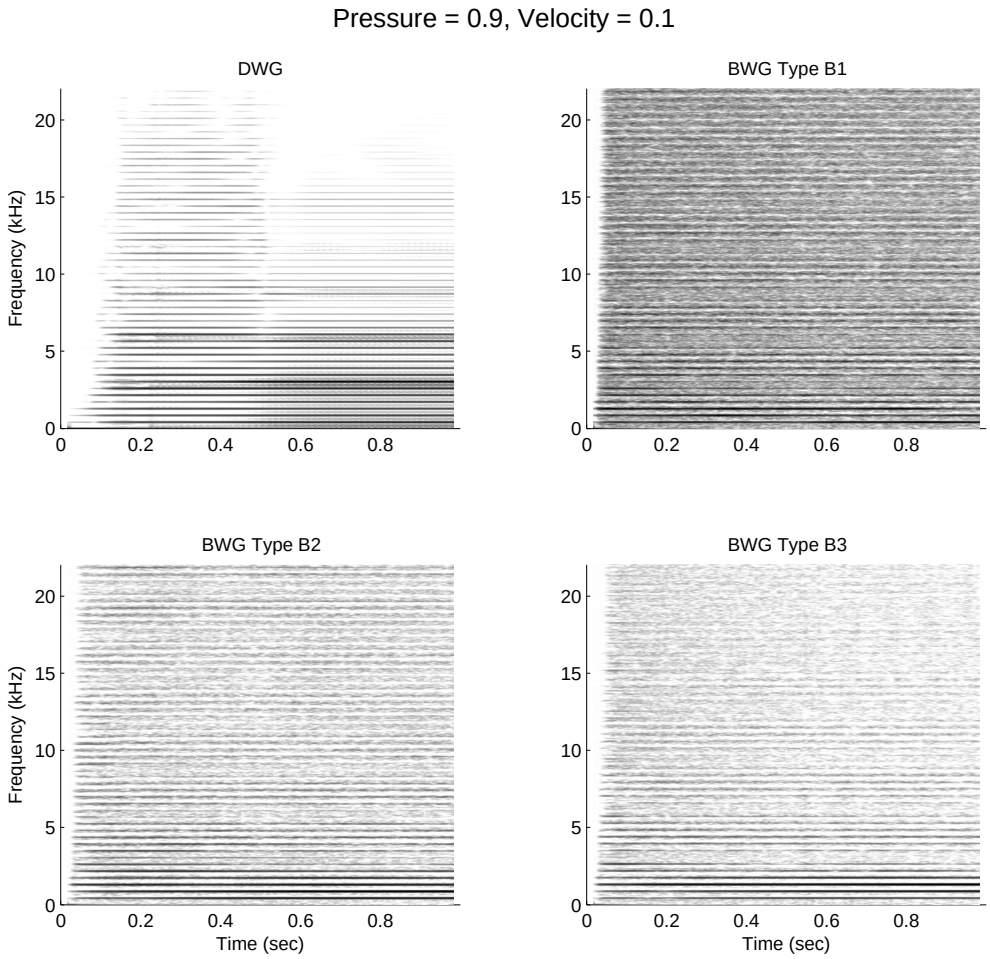


Figure C.32: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

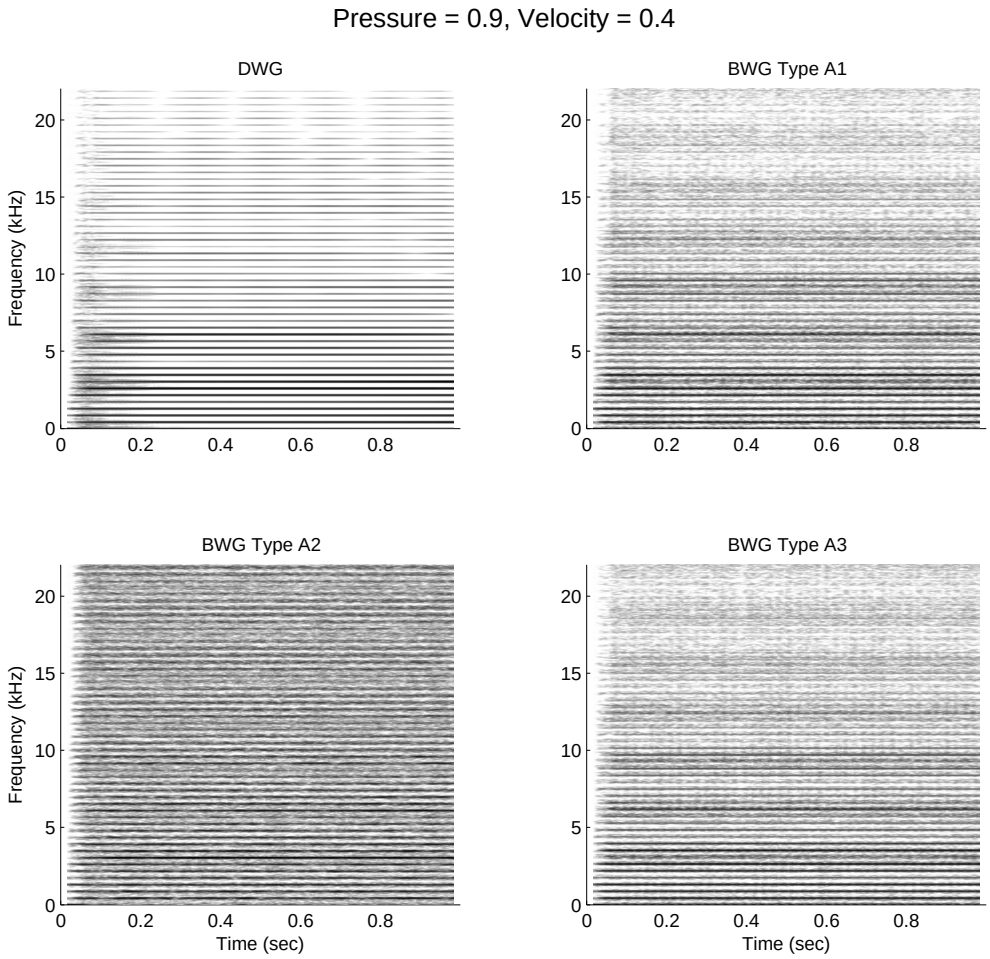


Figure C.33: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

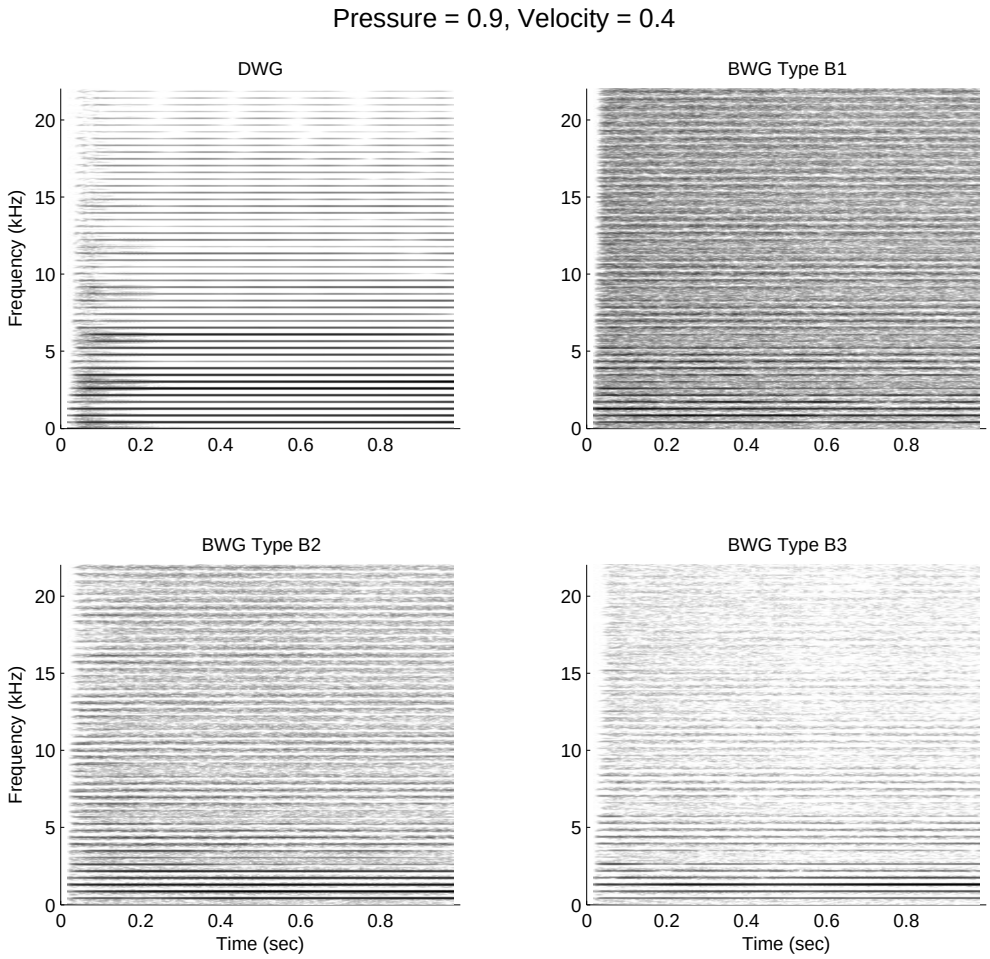


Figure C.34: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

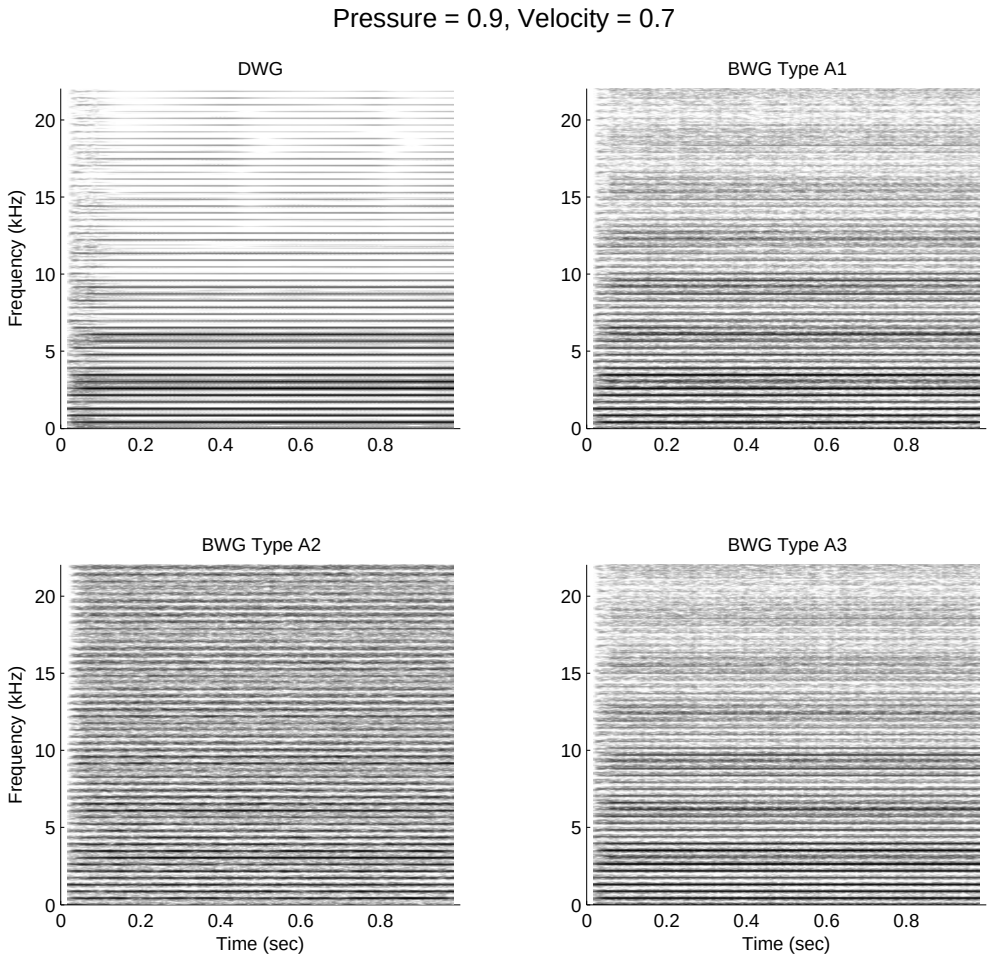


Figure C.35: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

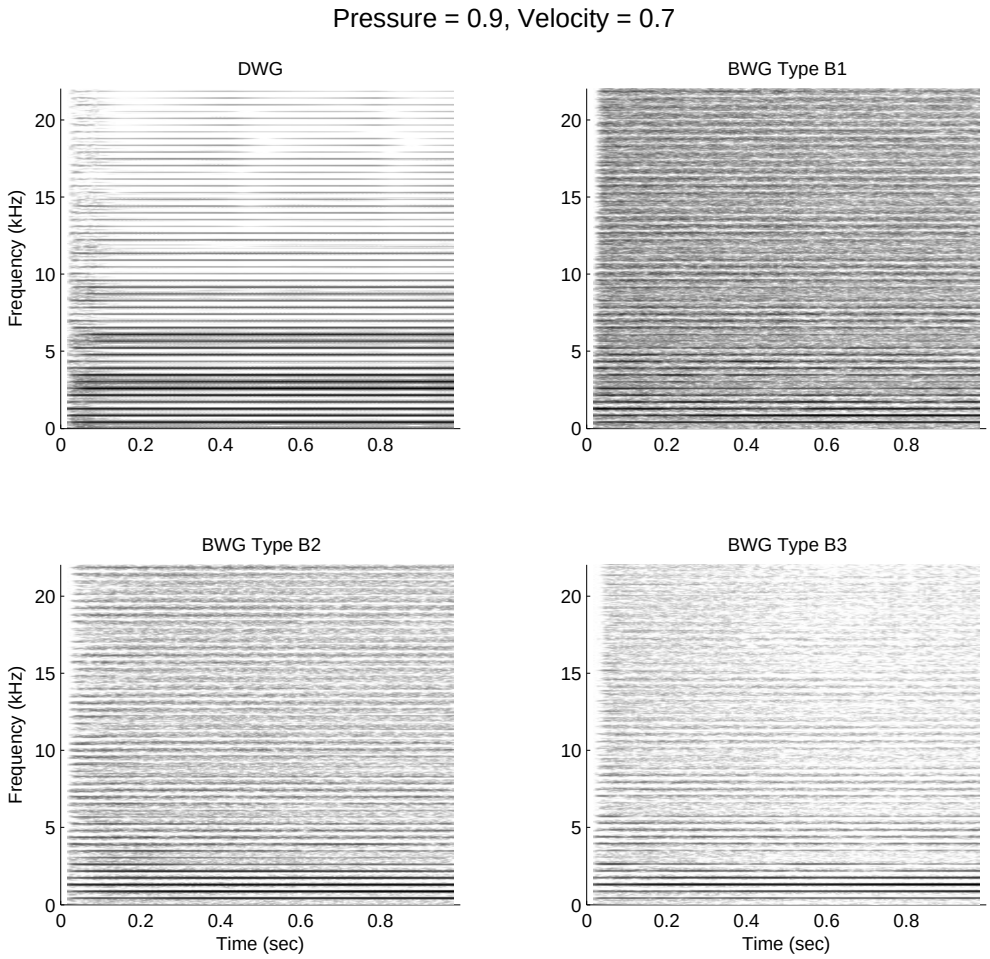


Figure C.36: Simulation of models of a bowed string with no stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

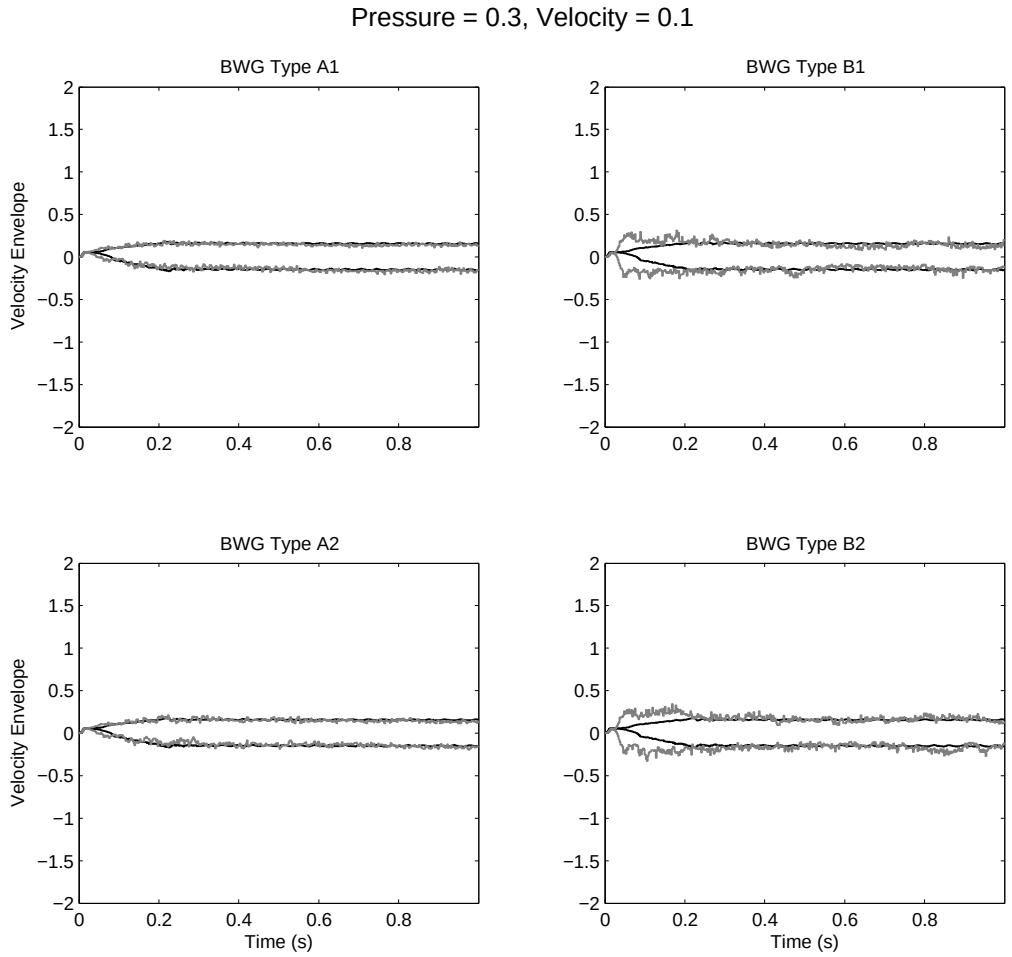


Figure C.37: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

C.2 Bowed String Tuned to 441 Hz with Stiffness

The following figures show the velocity amplitude envelope, steady-state oscillations, and spectrograms of various, comparably designed, BWG and DWG models. All models are of a string tuned to 441 Hz with stiffness, as described in Section 4.4.5.

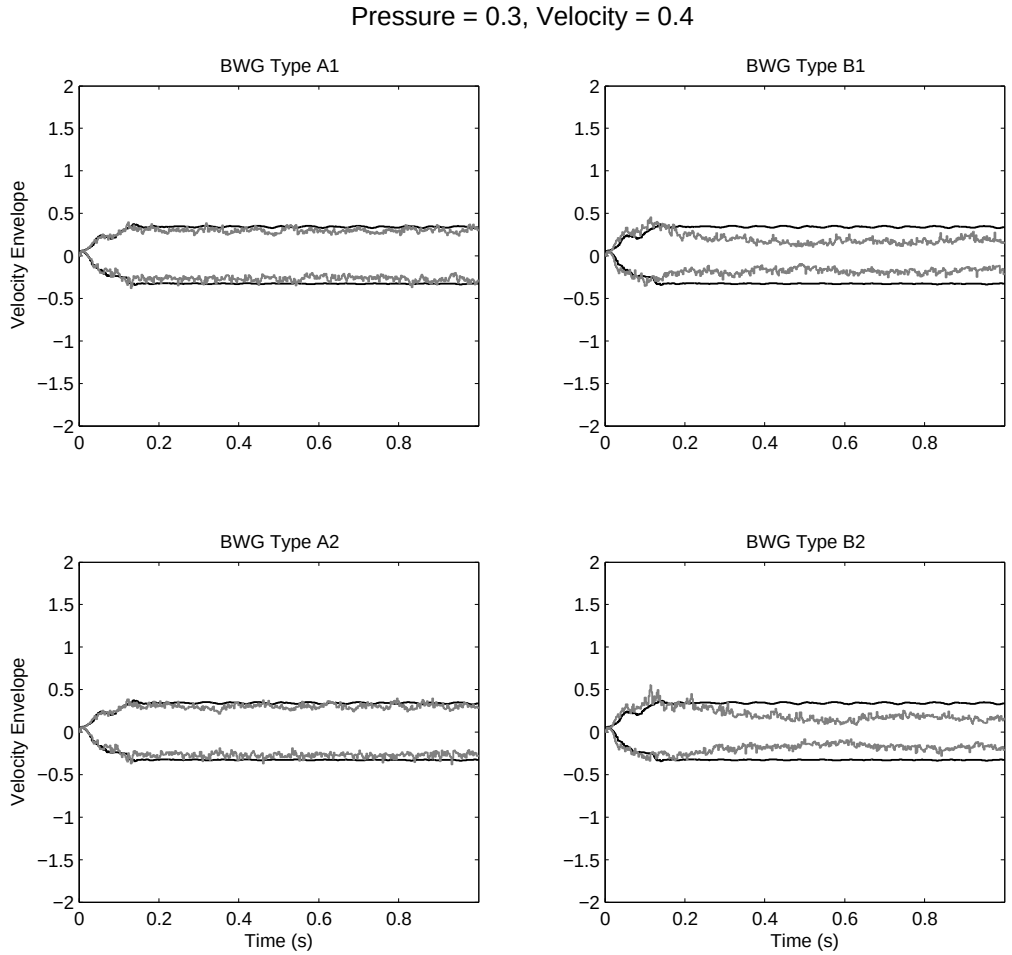


Figure C.38: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

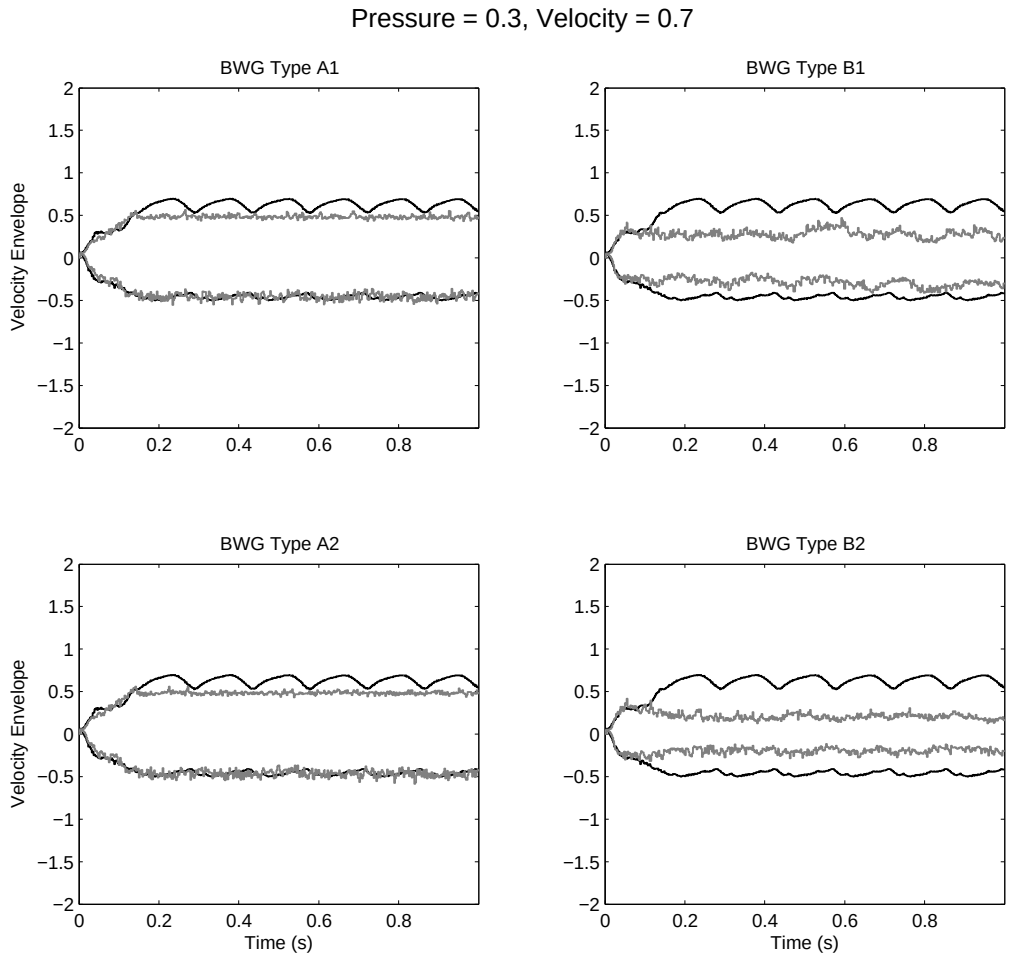


Figure C.39: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

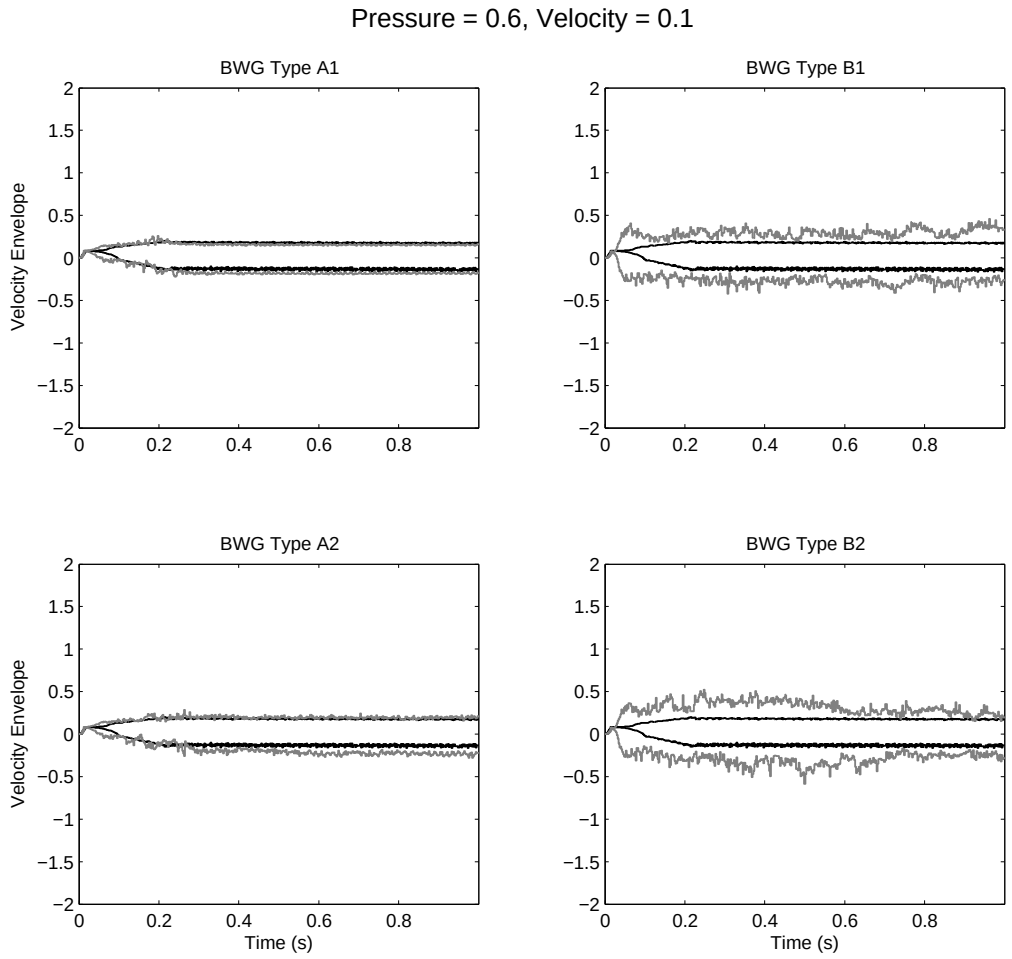


Figure C.40: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

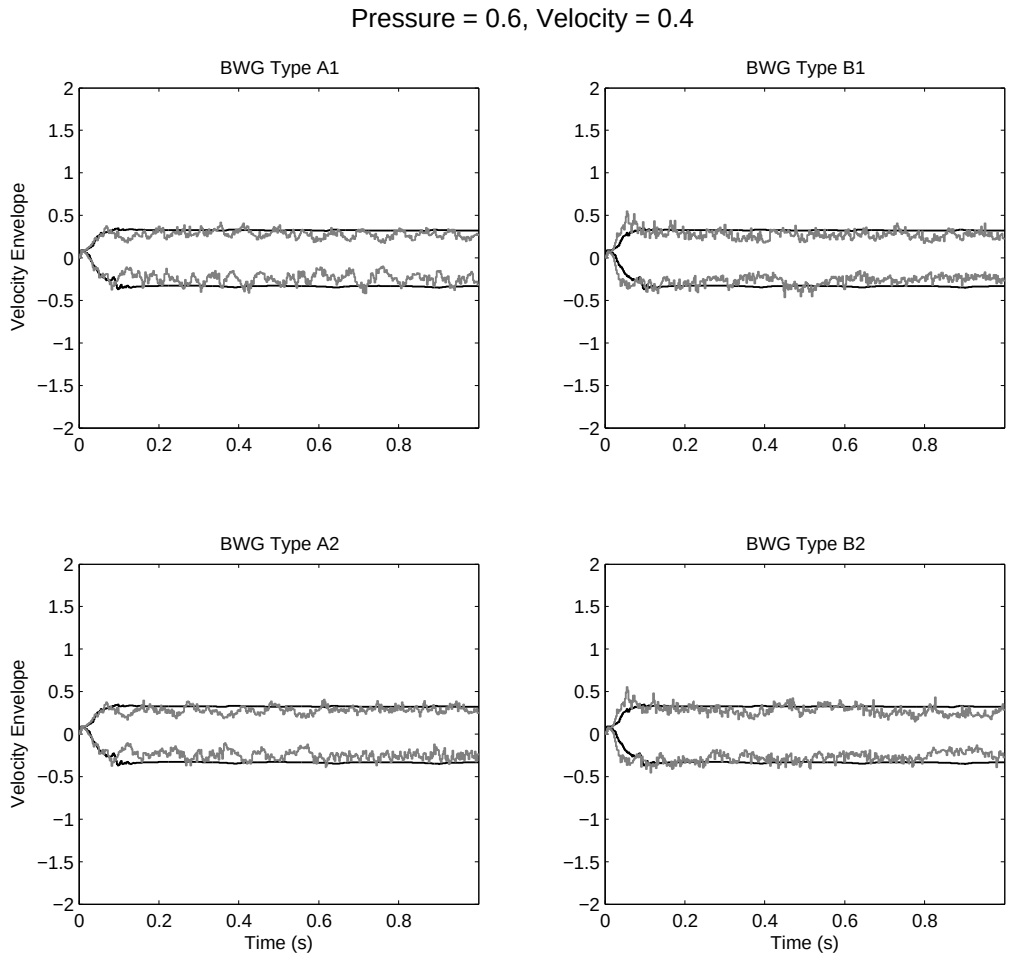


Figure C.41: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

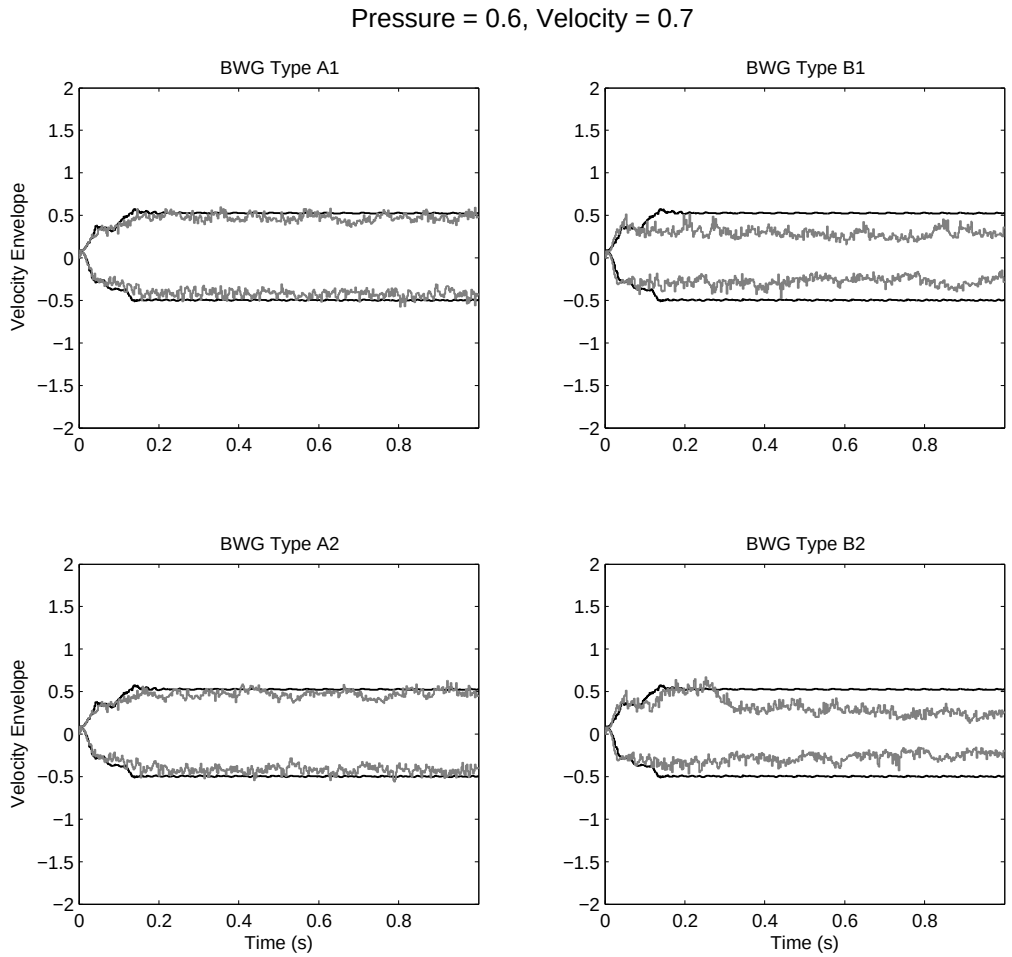


Figure C.42: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

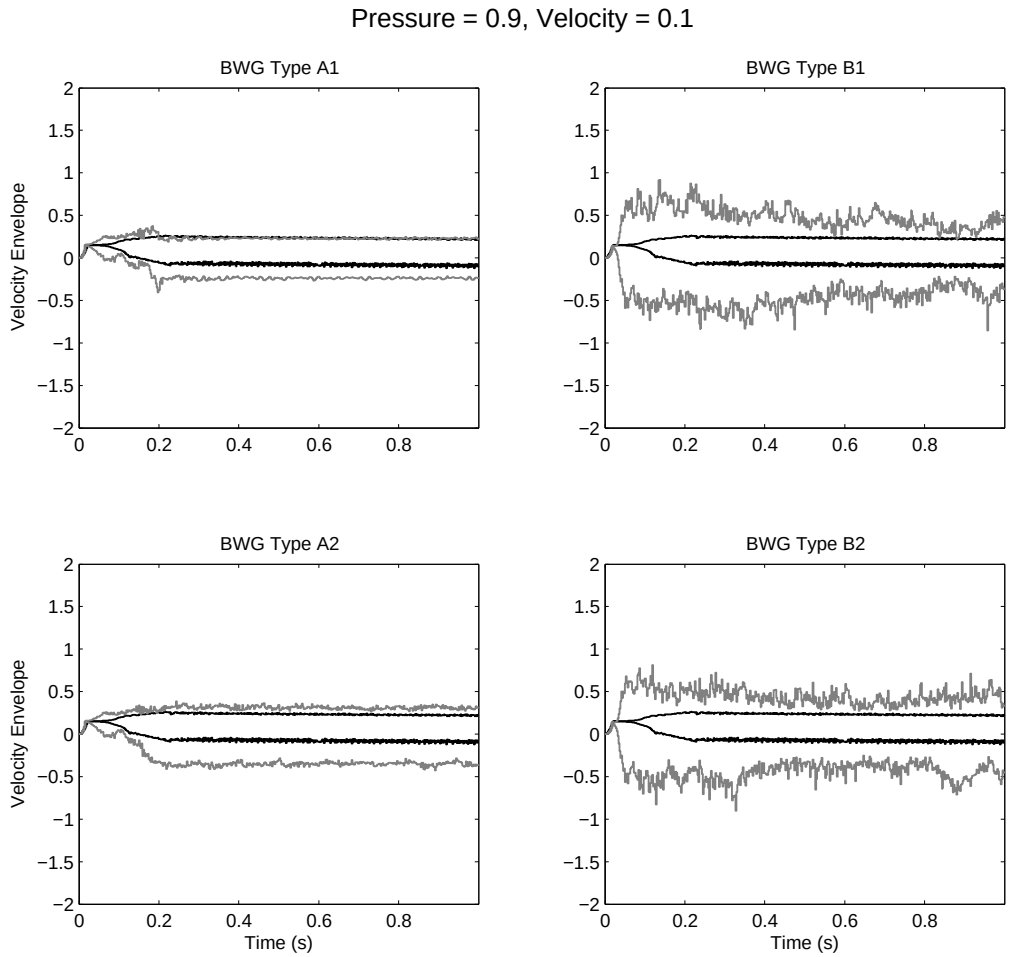


Figure C.43: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

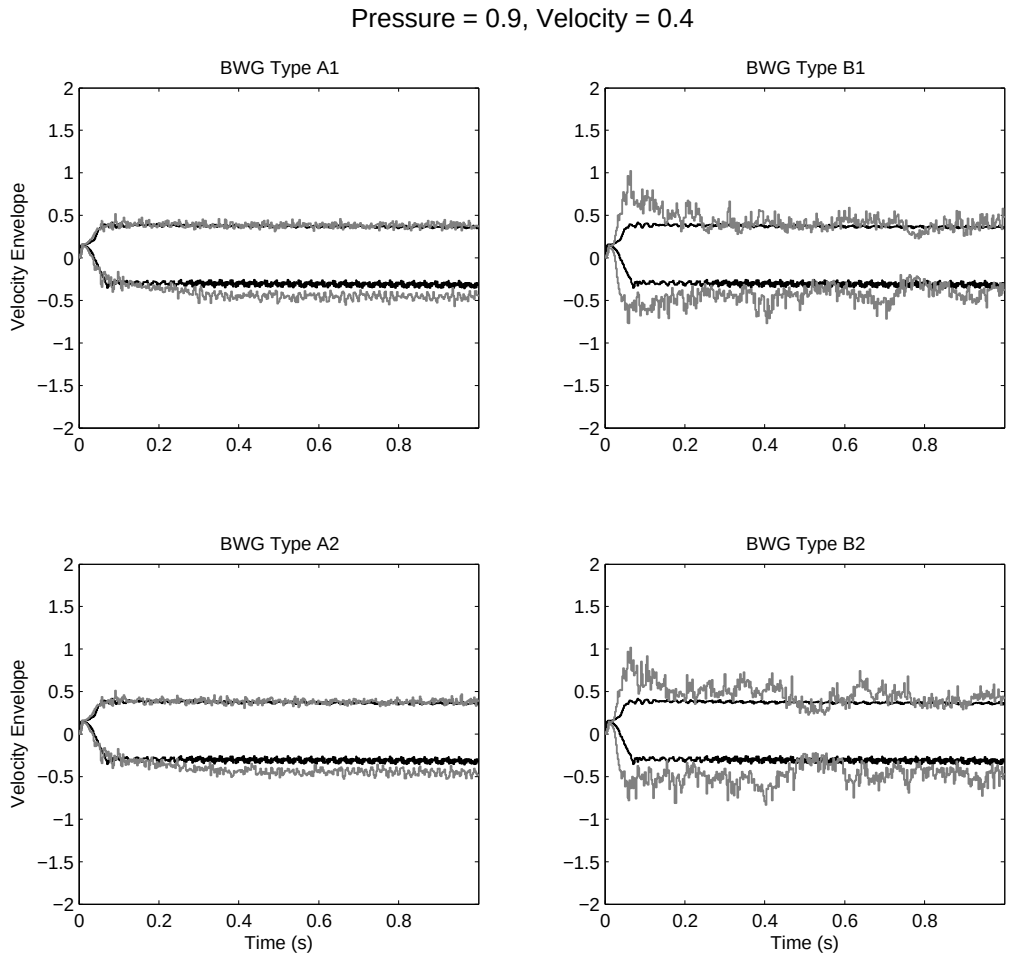


Figure C.44: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

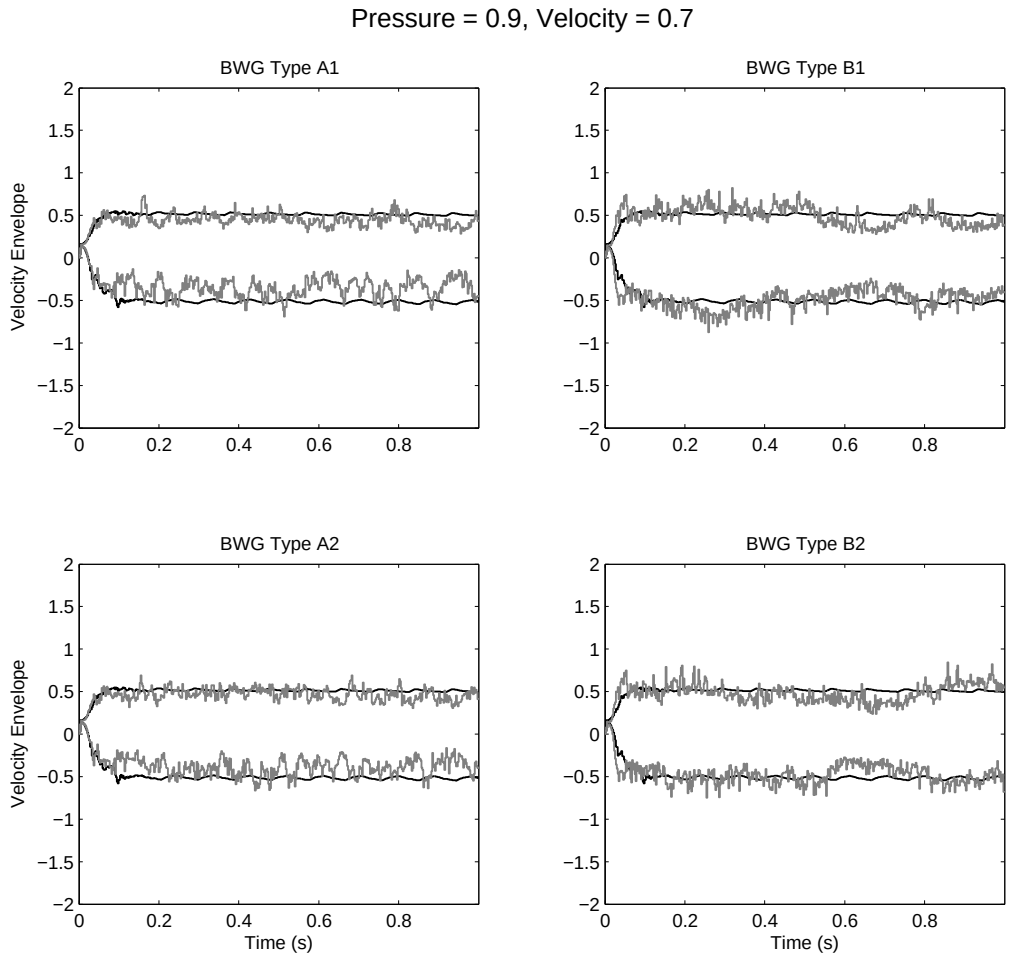


Figure C.45: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The velocity envelope of the output of various BWG models is shown, compared to that of a comparable DWG model. The DWG model's velocity envelope is shown in black, and each BWG velocity envelope is shown in gray.

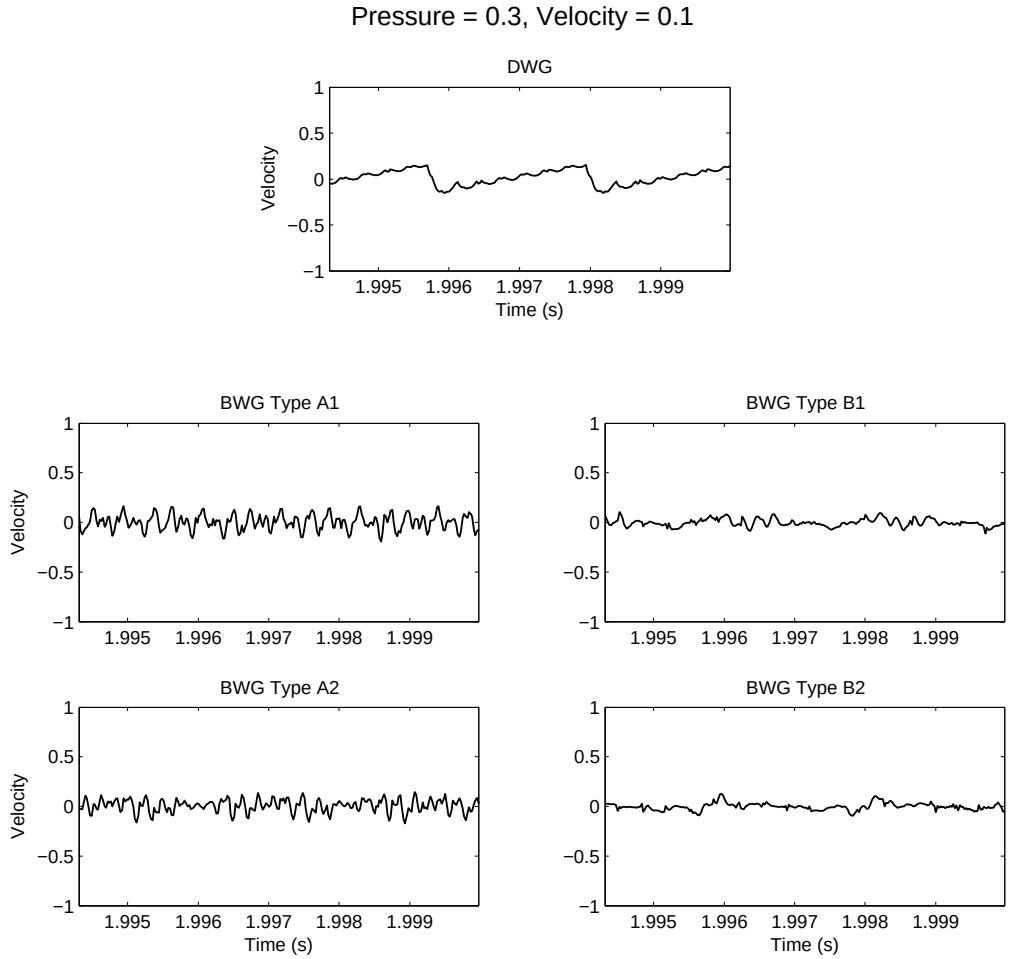


Figure C.46: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

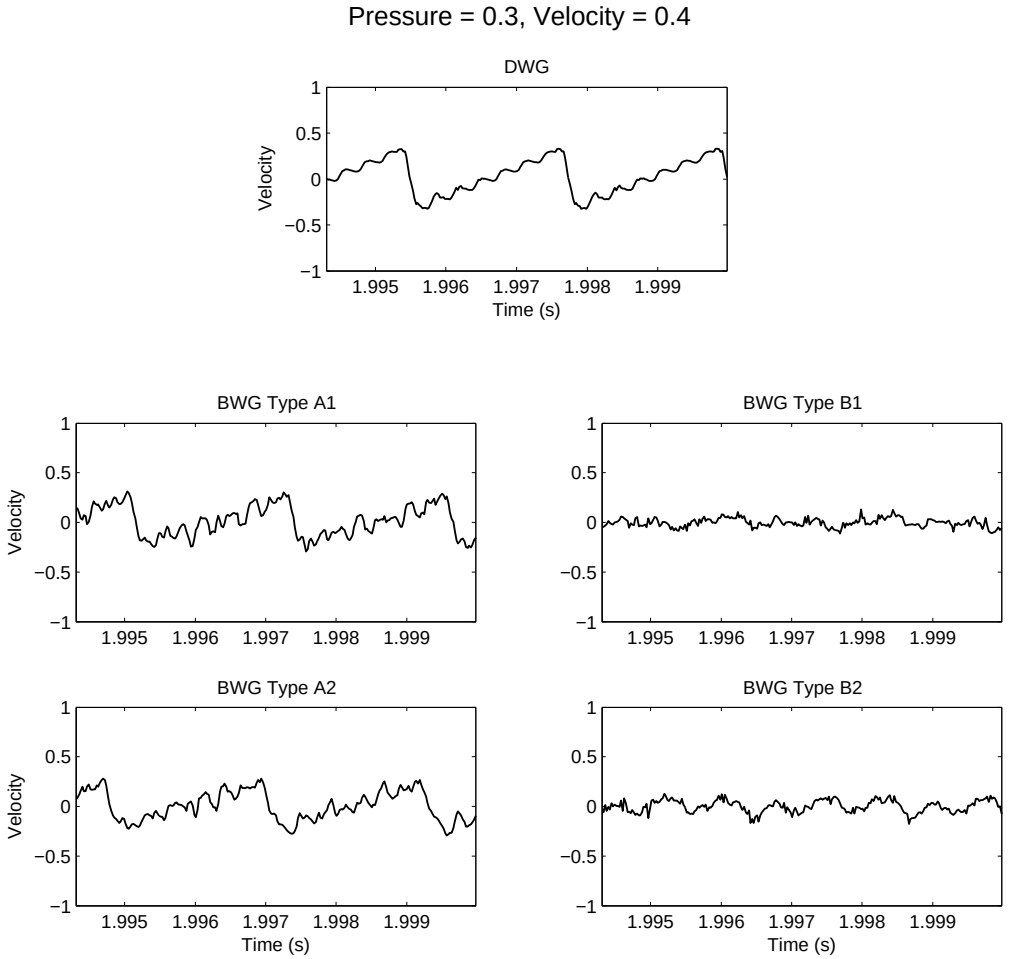


Figure C.47: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

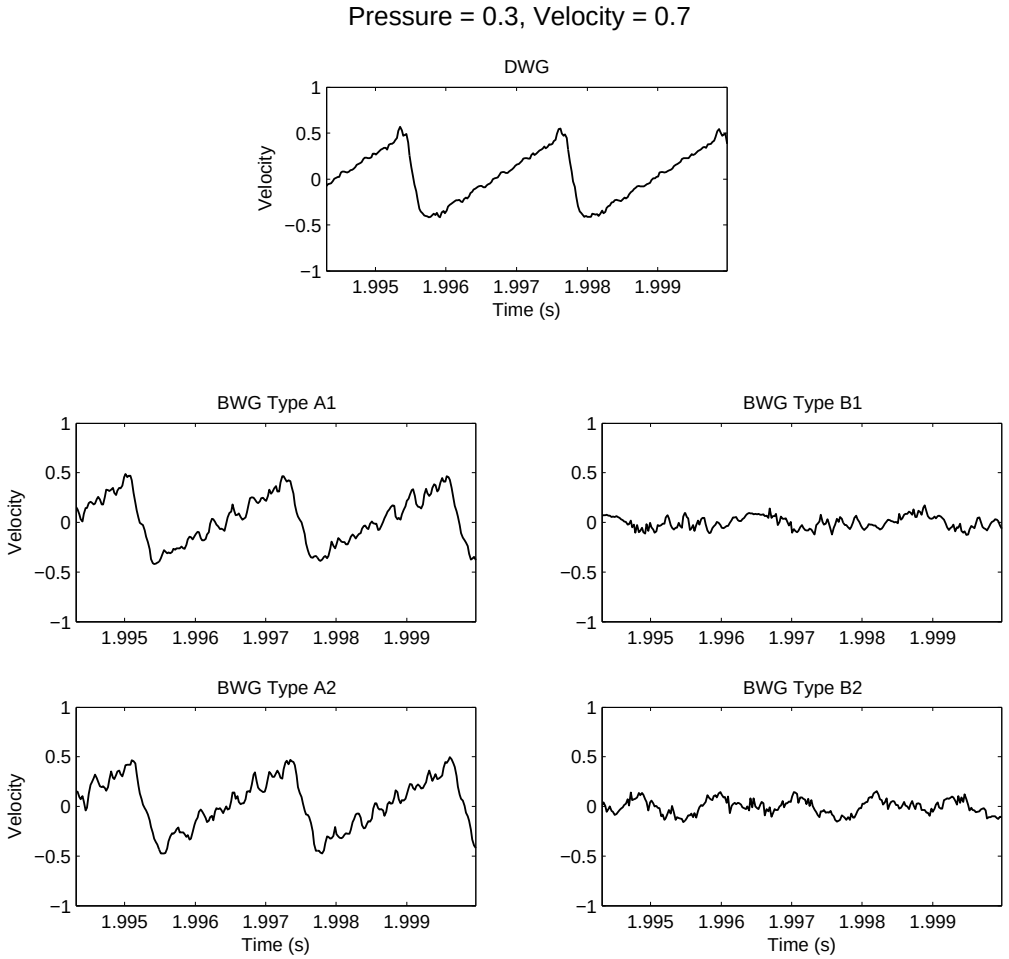


Figure C.48: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

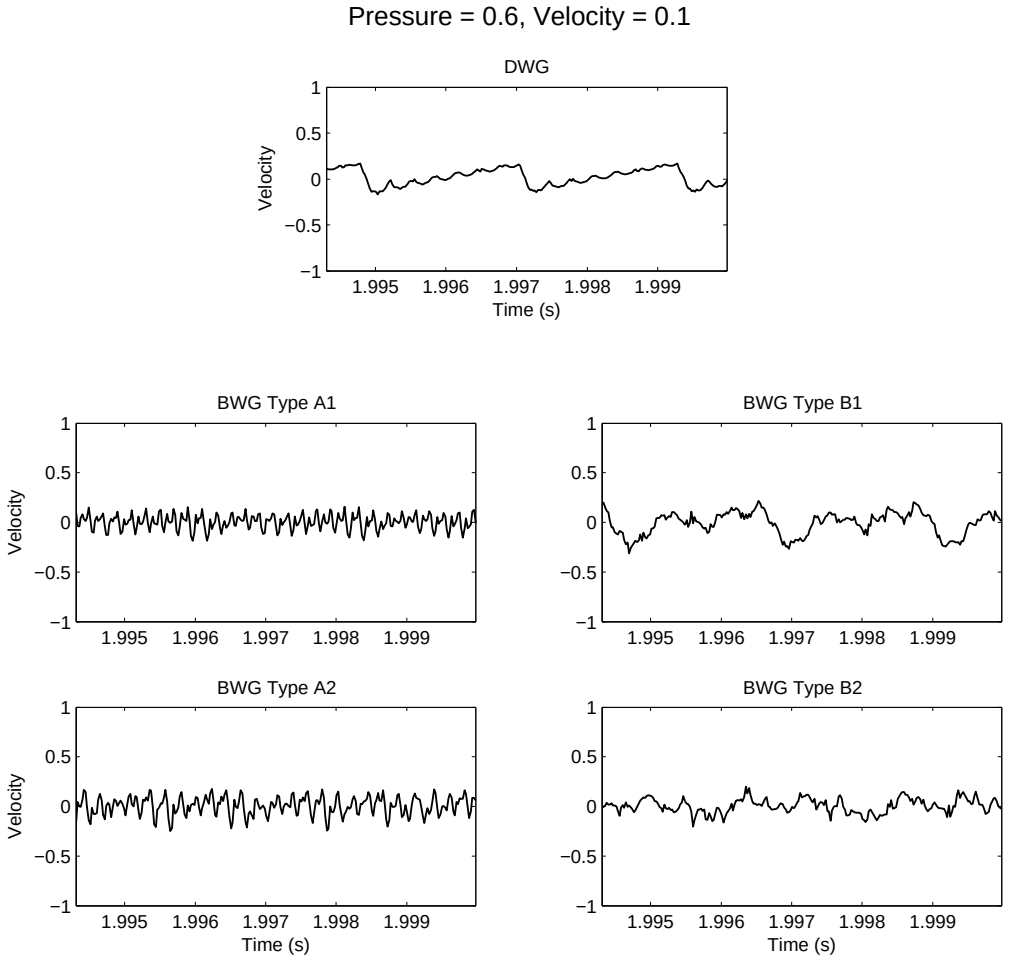


Figure C.49: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

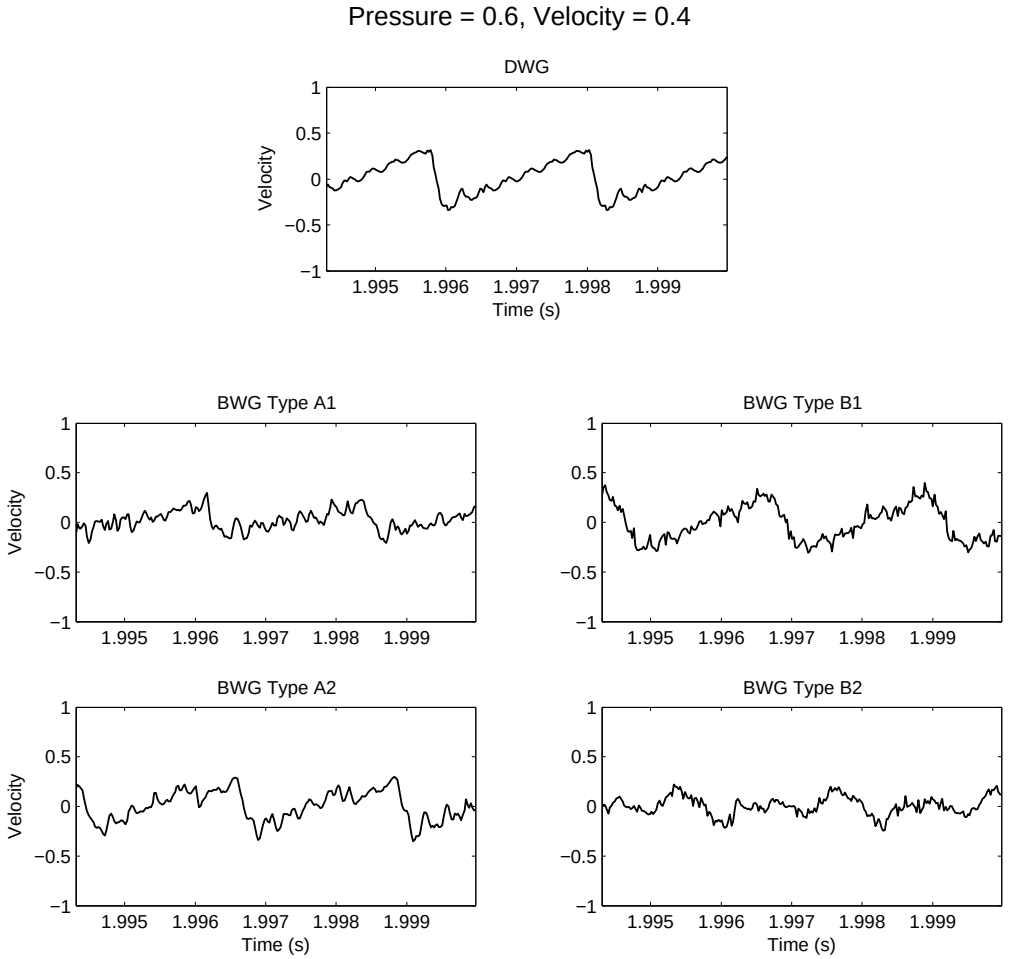


Figure C.50: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

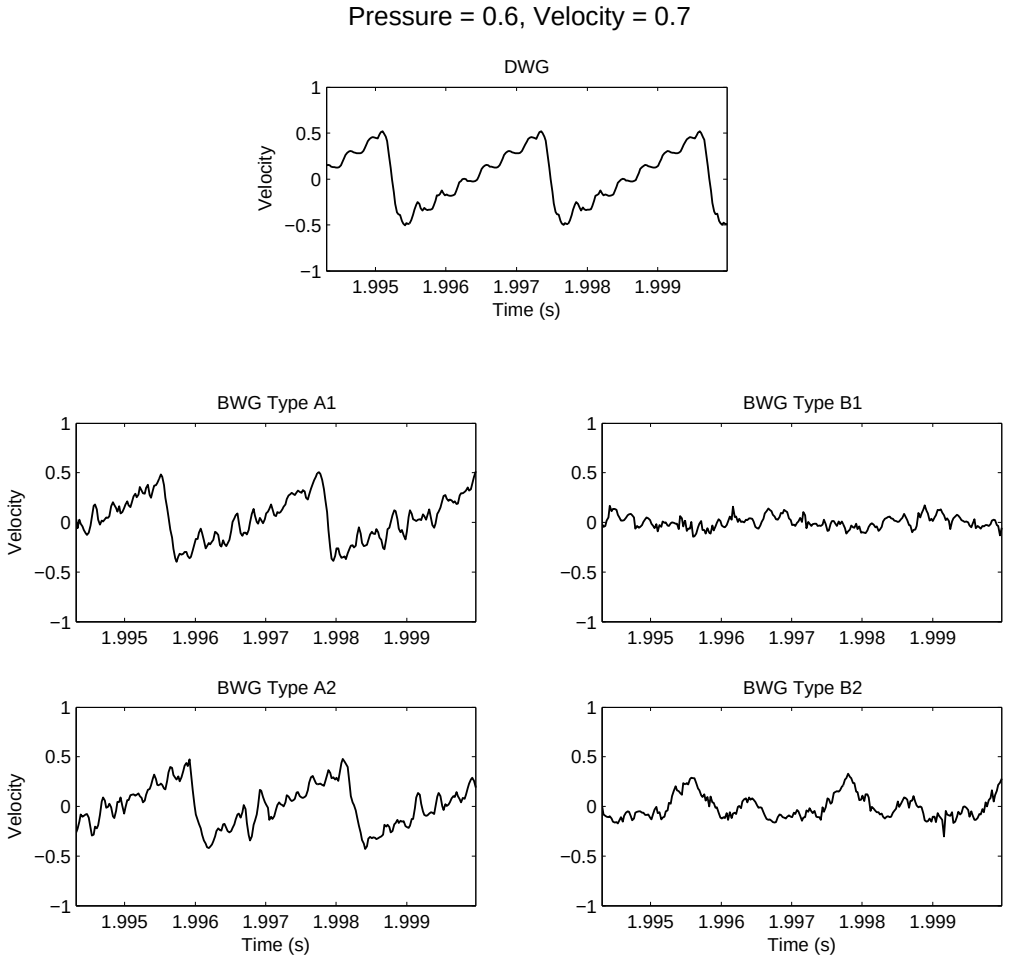


Figure C.51: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

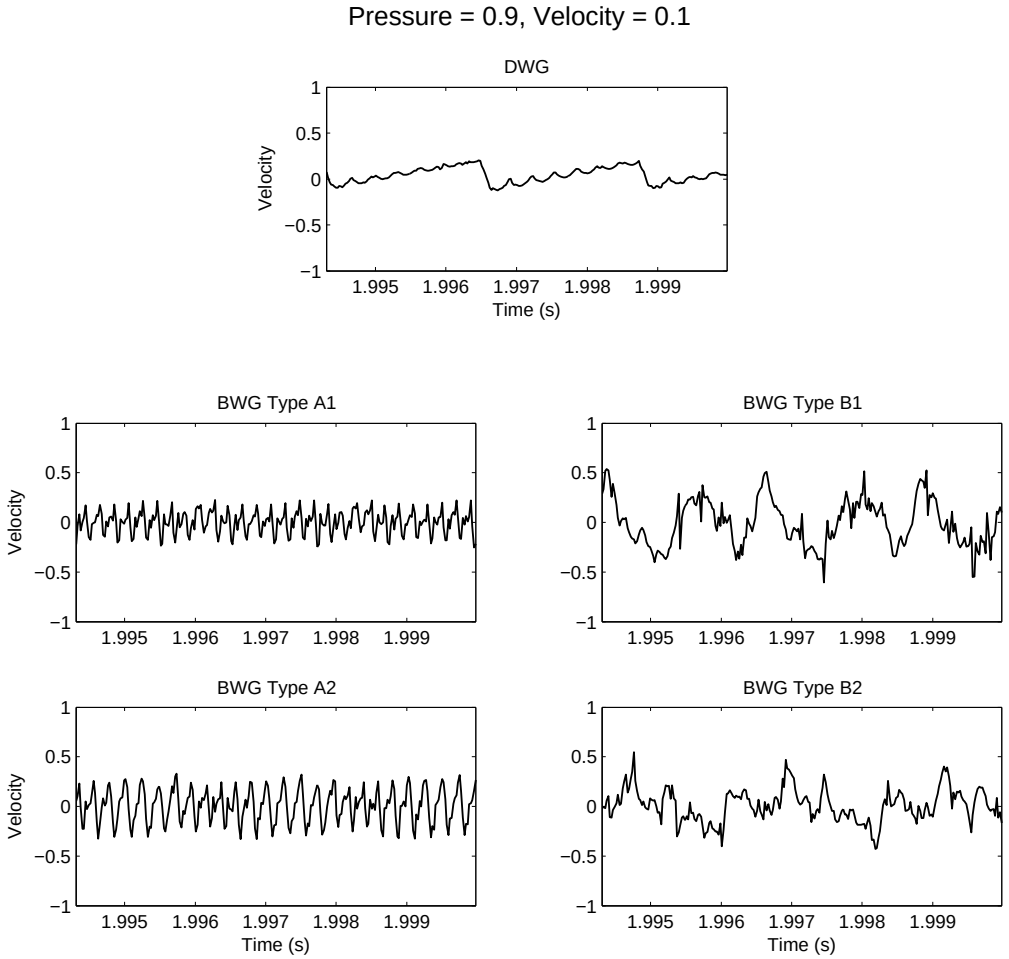


Figure C.52: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

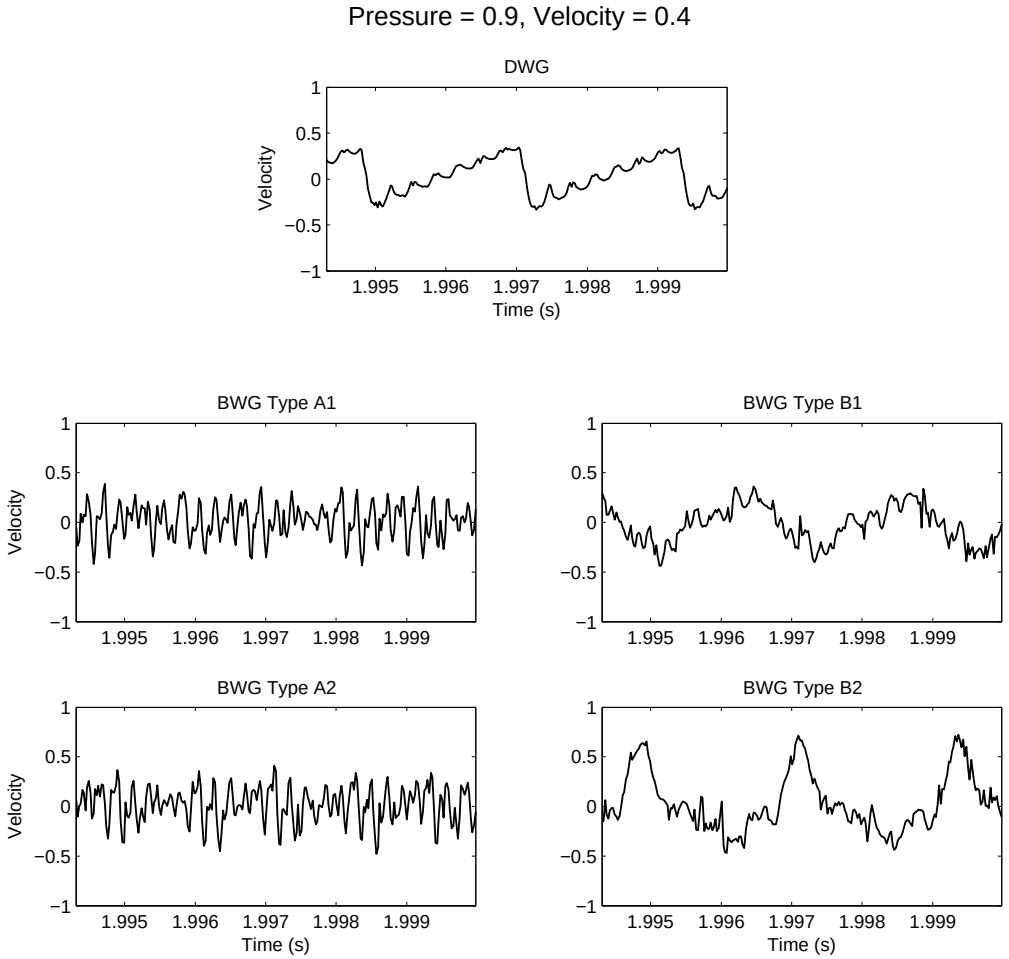


Figure C.53: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

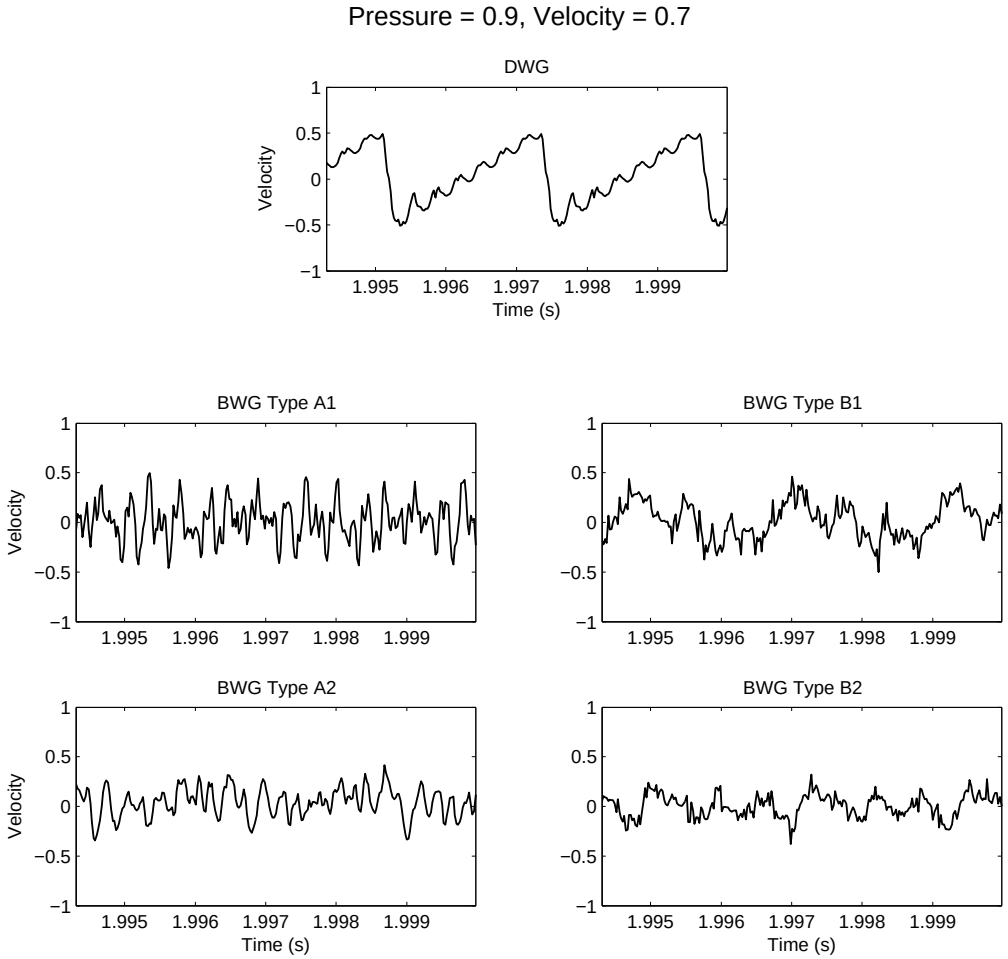


Figure C.54: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The steady-state velocity output of various BWG models is shown, compared to that of a comparable DWG model.

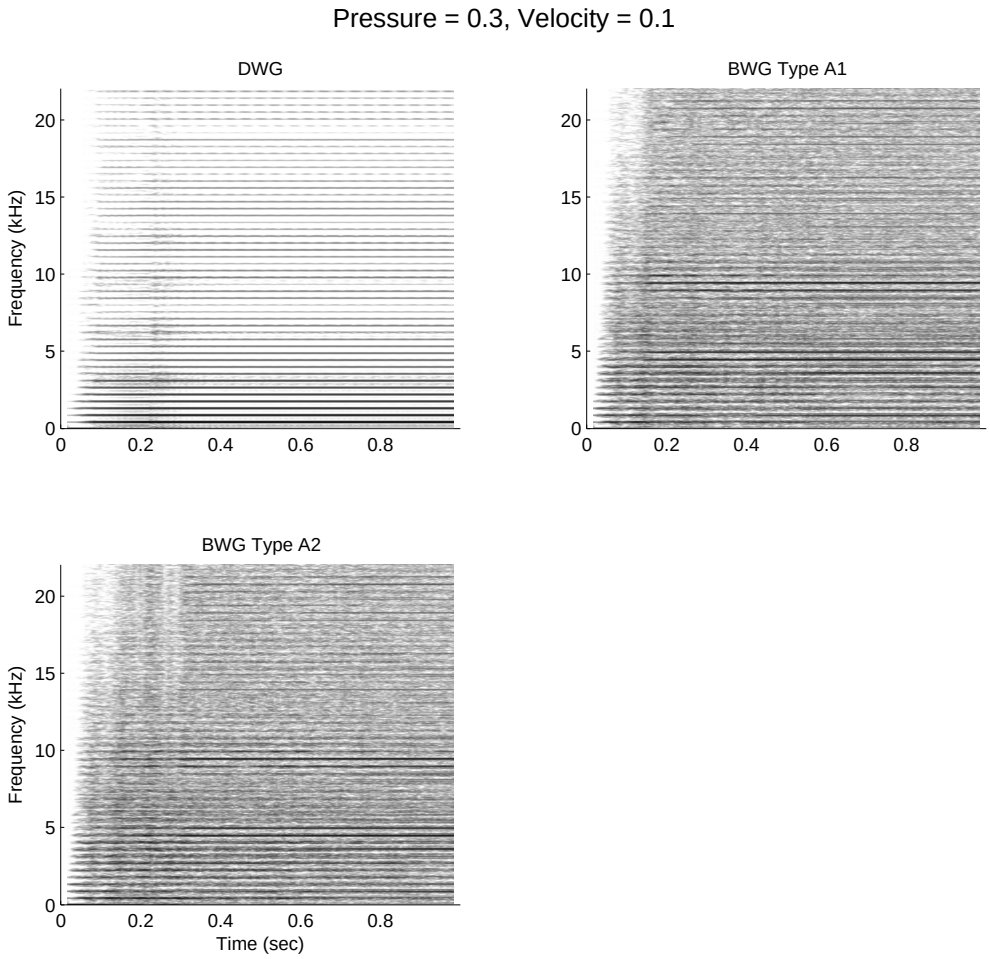


Figure C.55: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

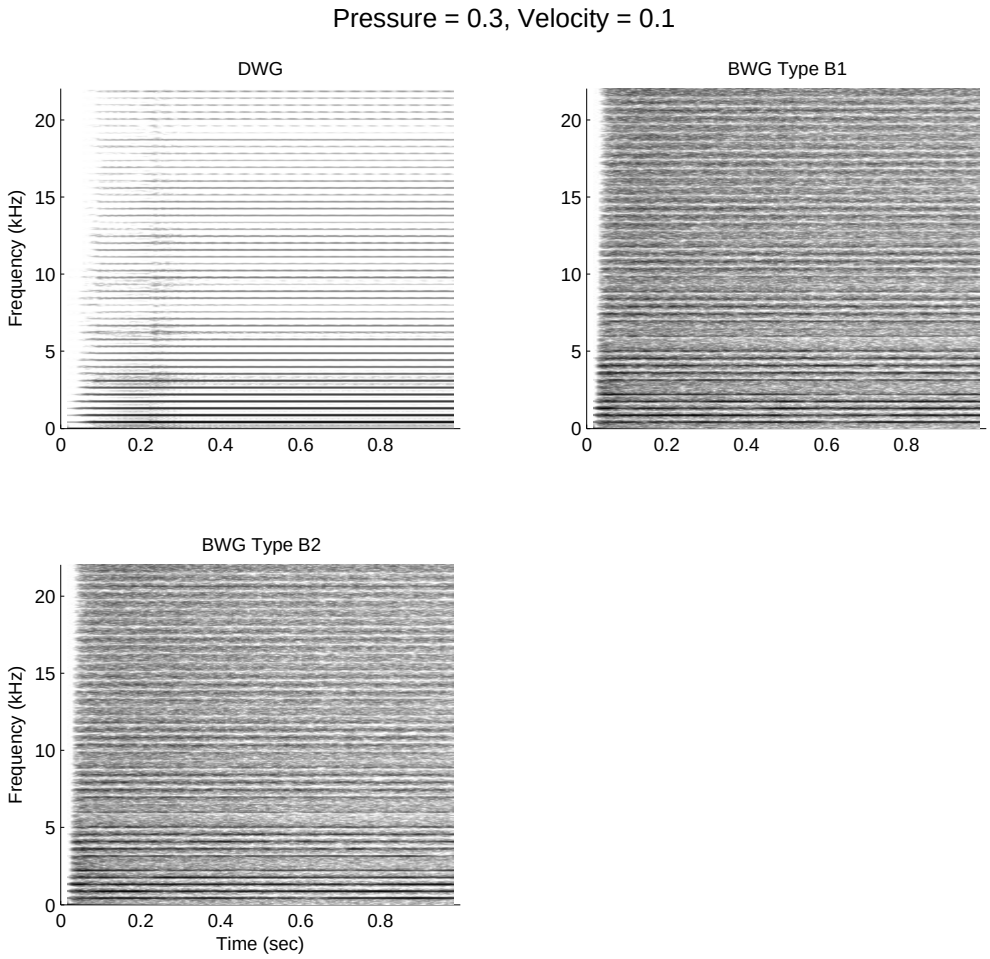


Figure C.56: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

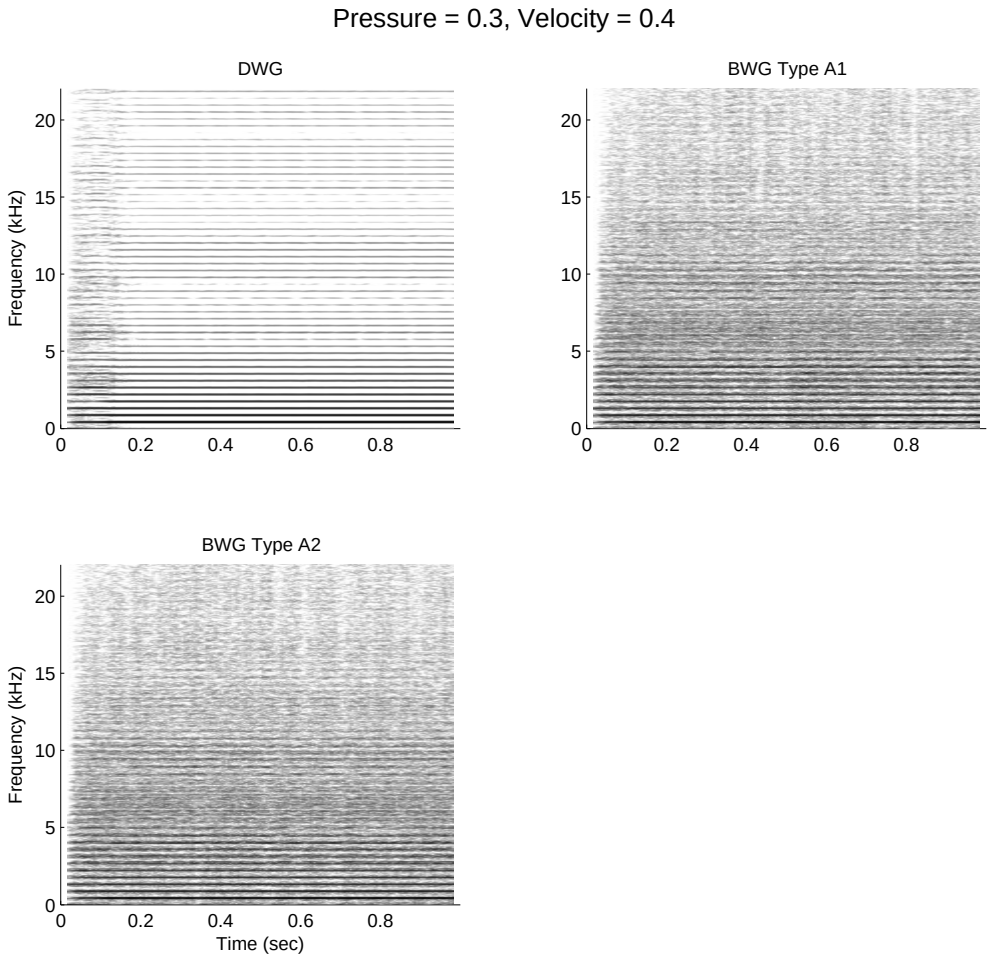


Figure C.57: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

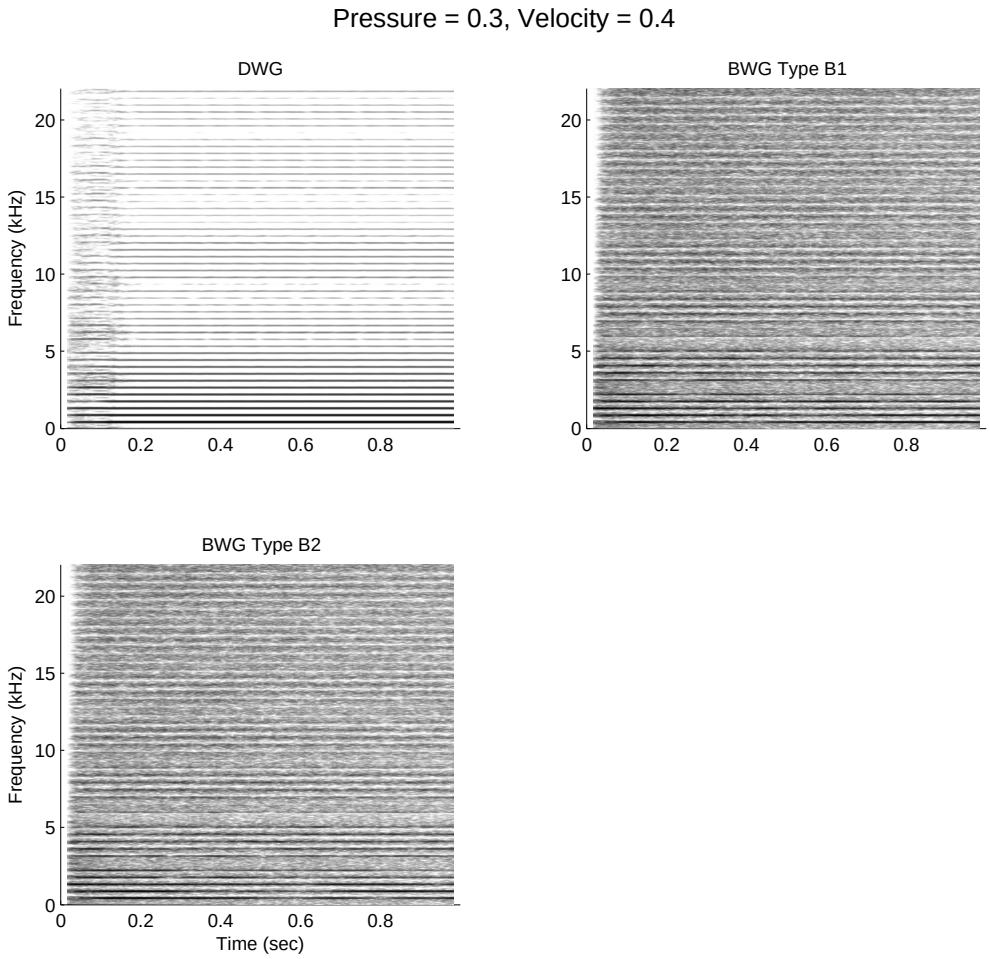


Figure C.58: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

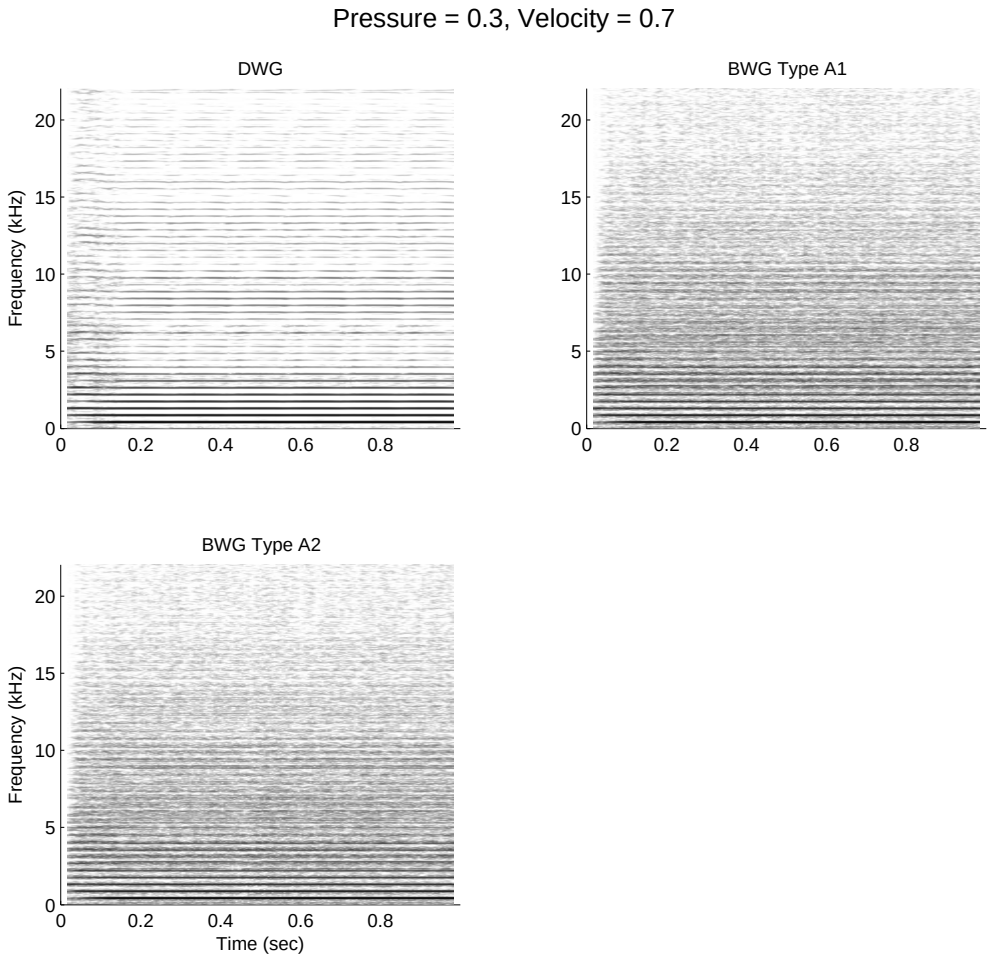


Figure C.59: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

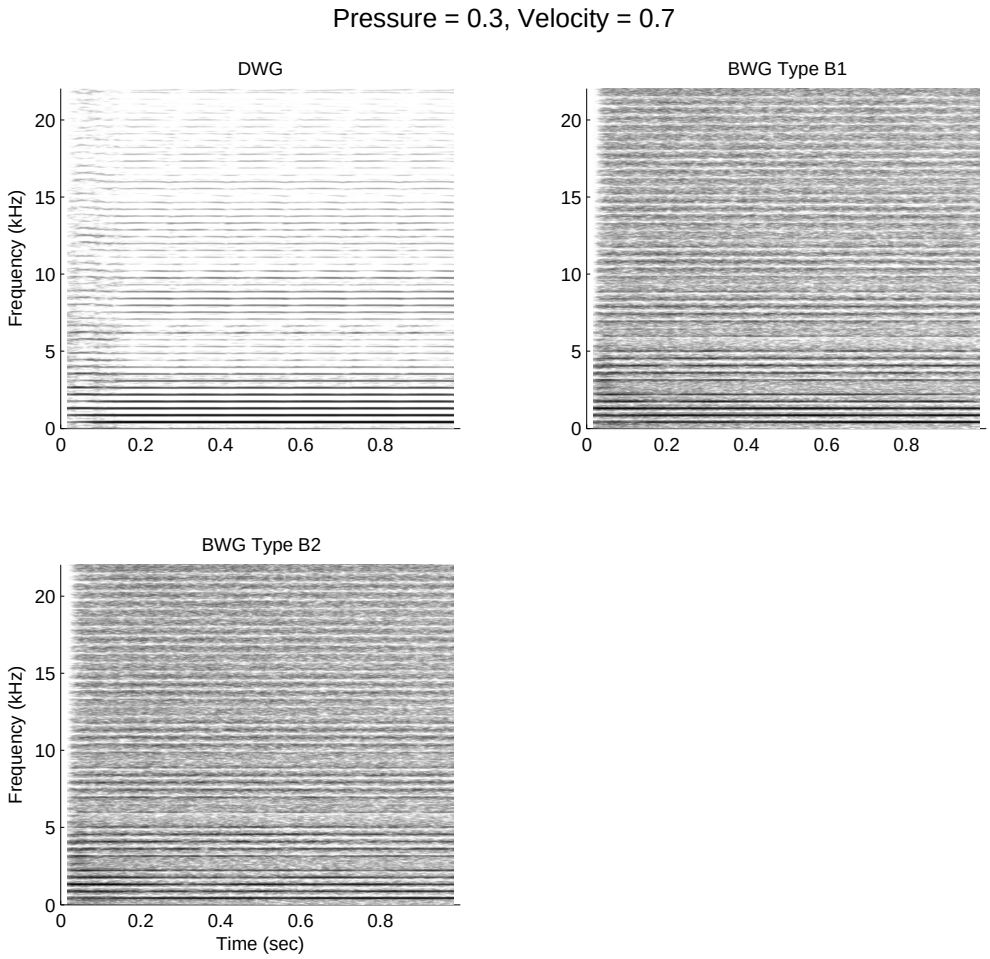


Figure C.60: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

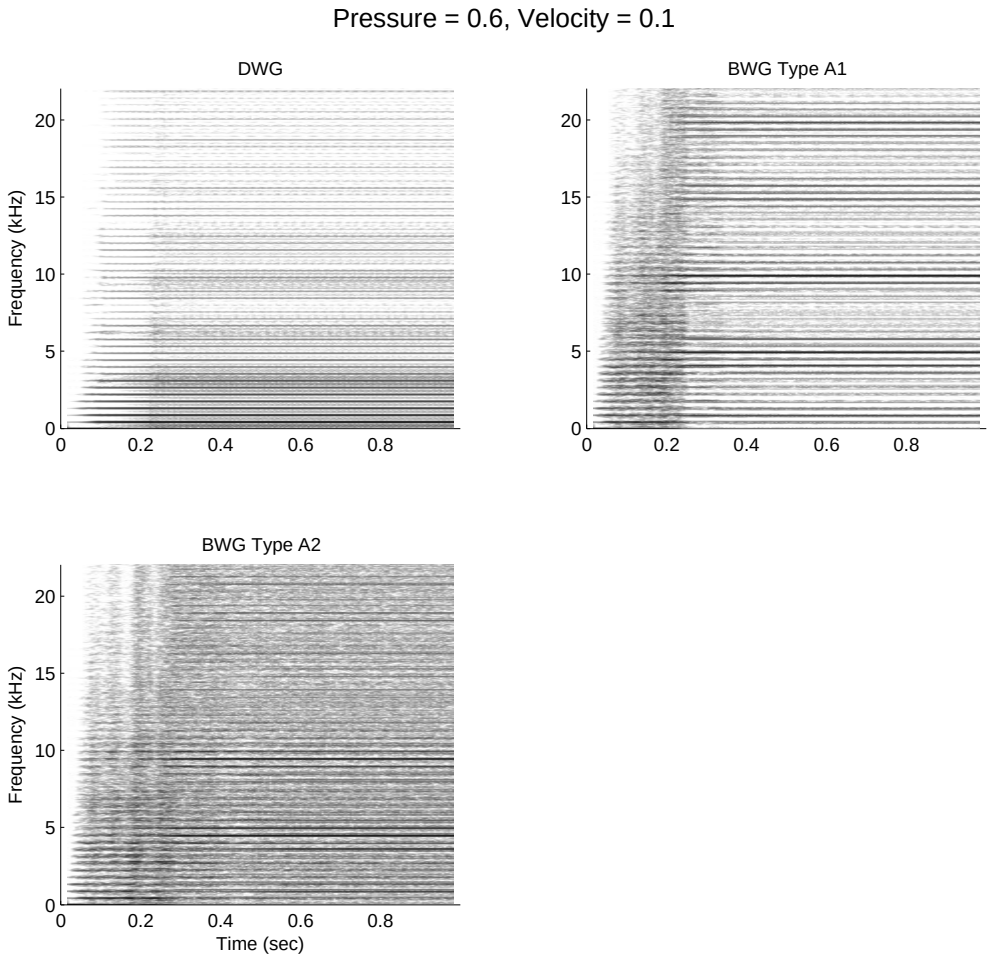


Figure C.61: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

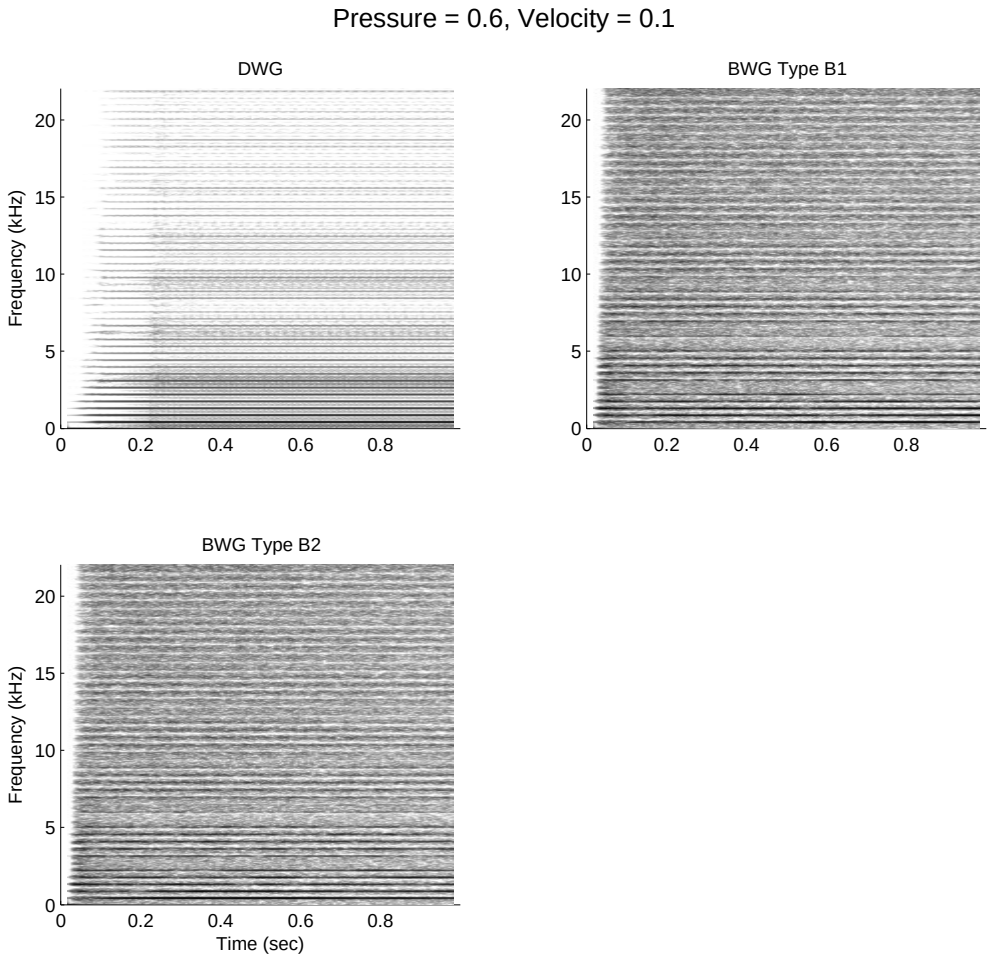


Figure C.62: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

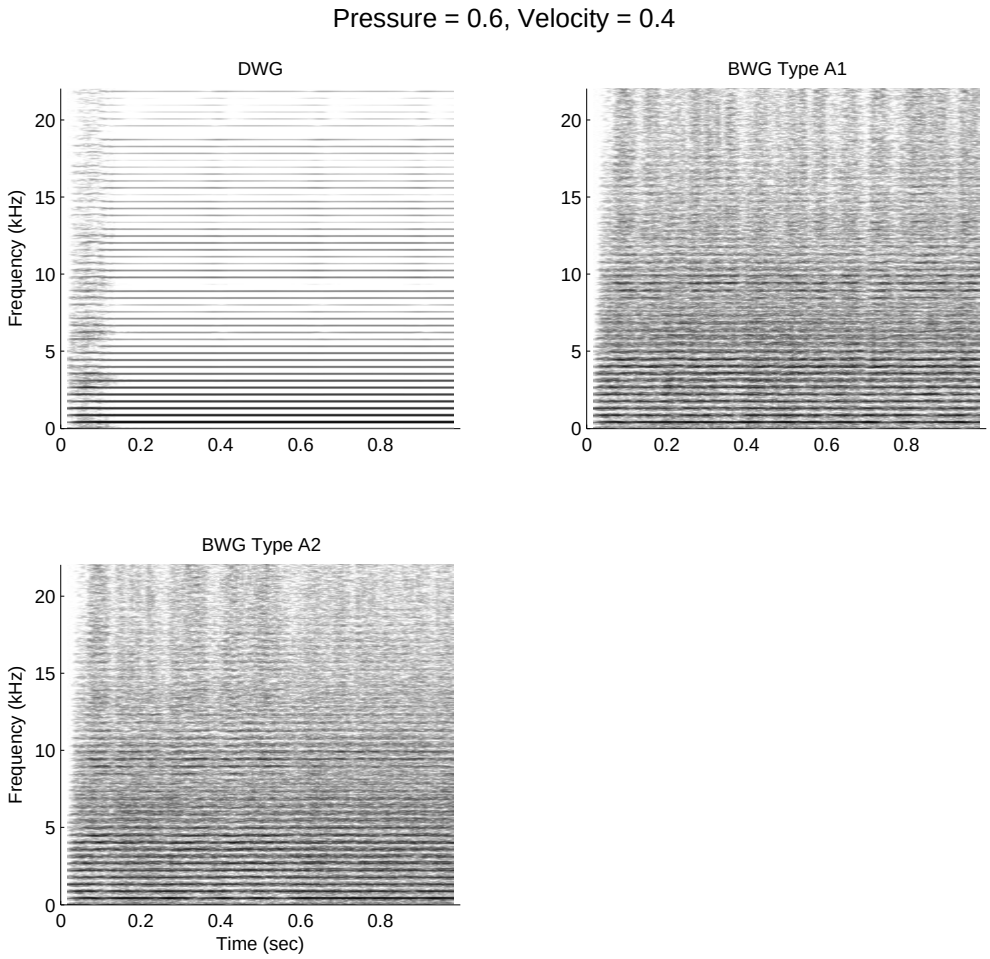


Figure C.63: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

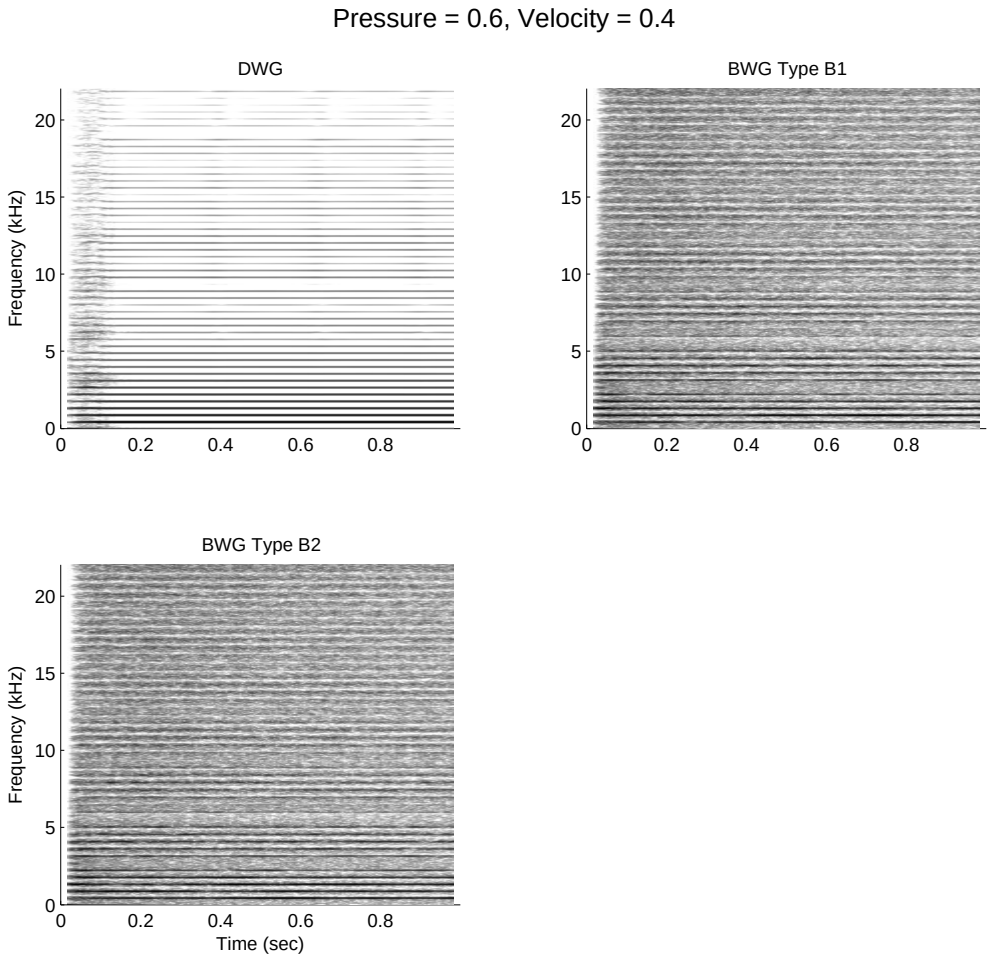


Figure C.64: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

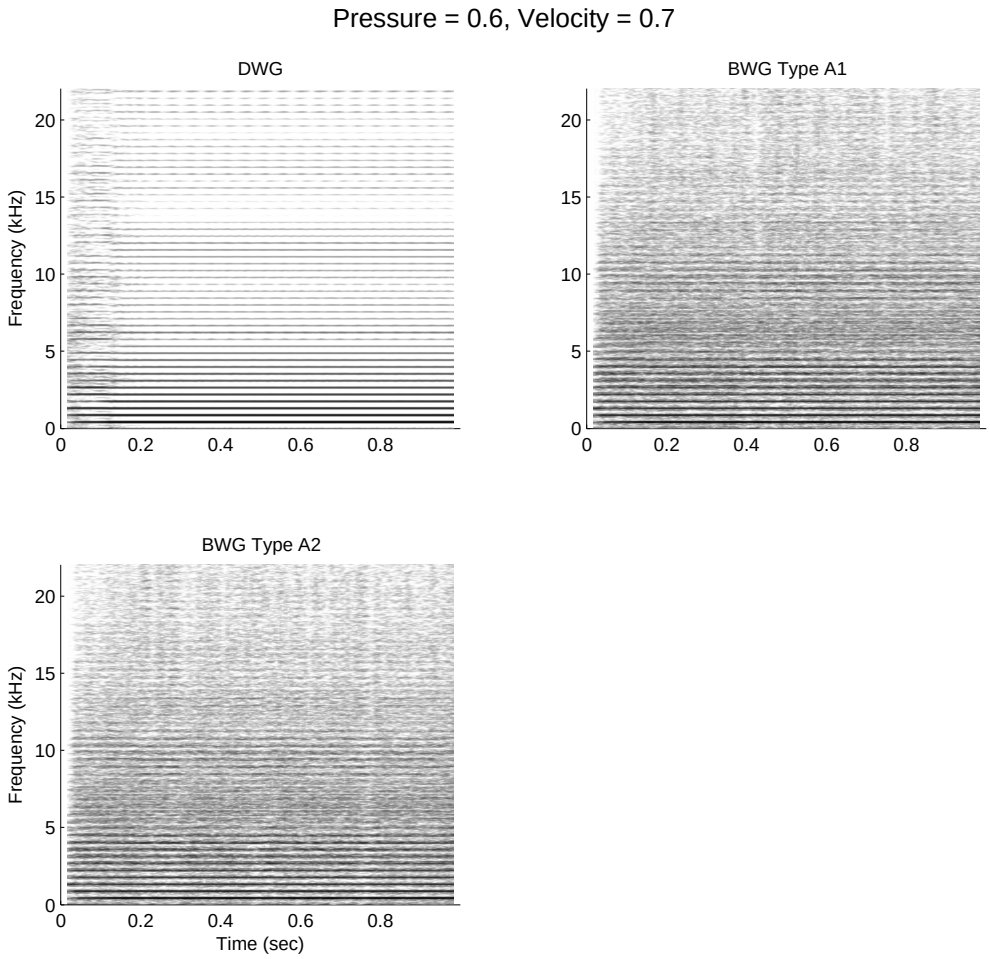


Figure C.65: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

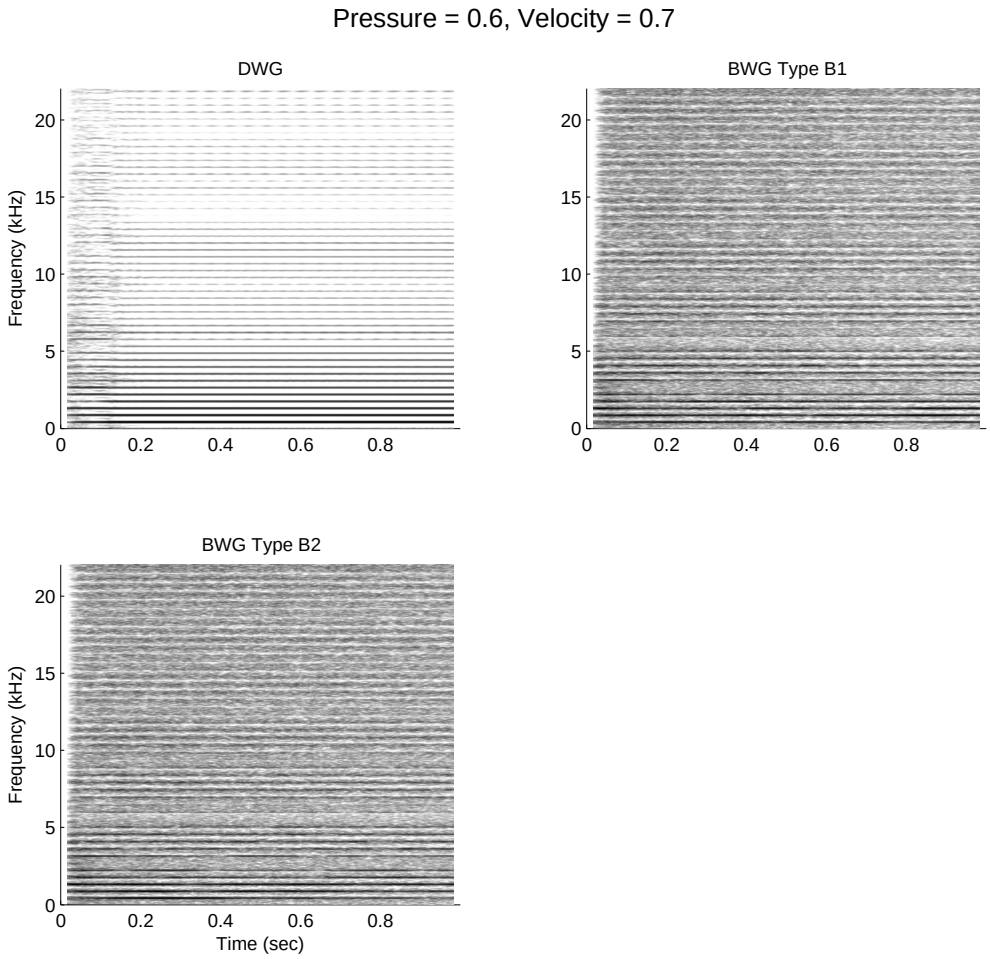


Figure C.66: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

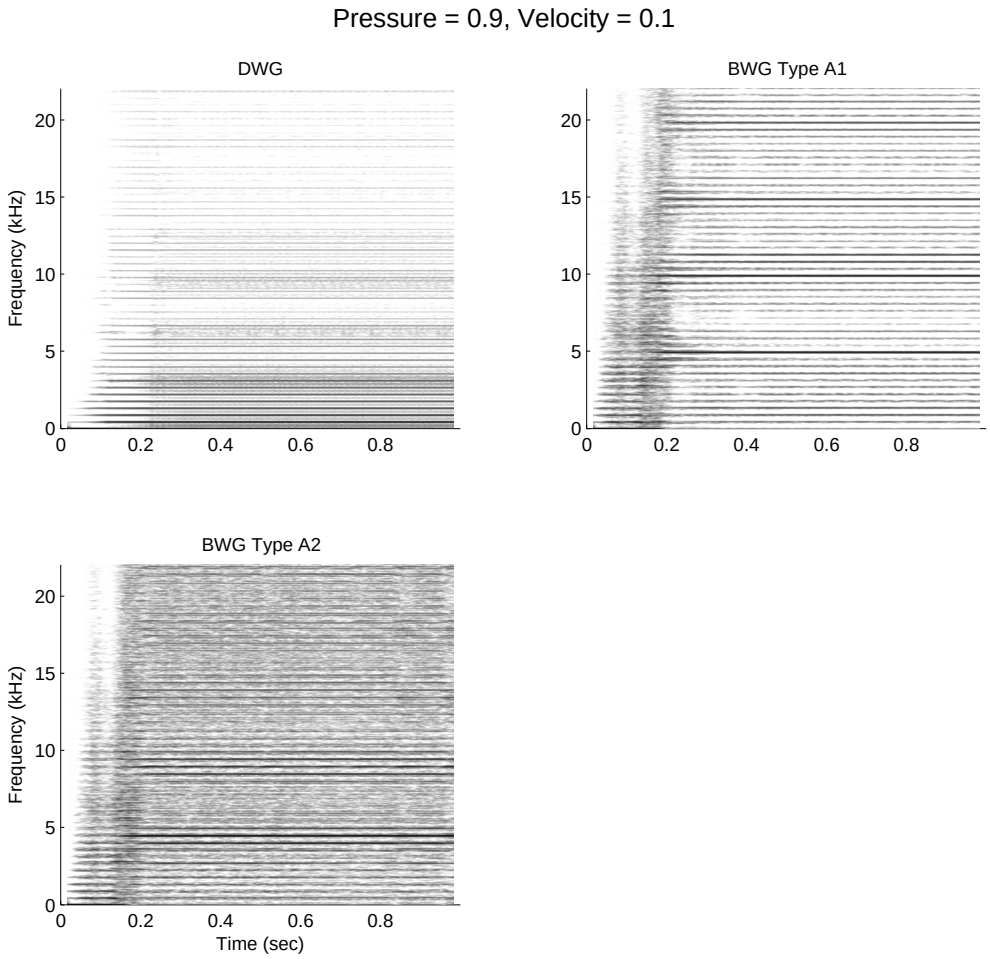


Figure C.67: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

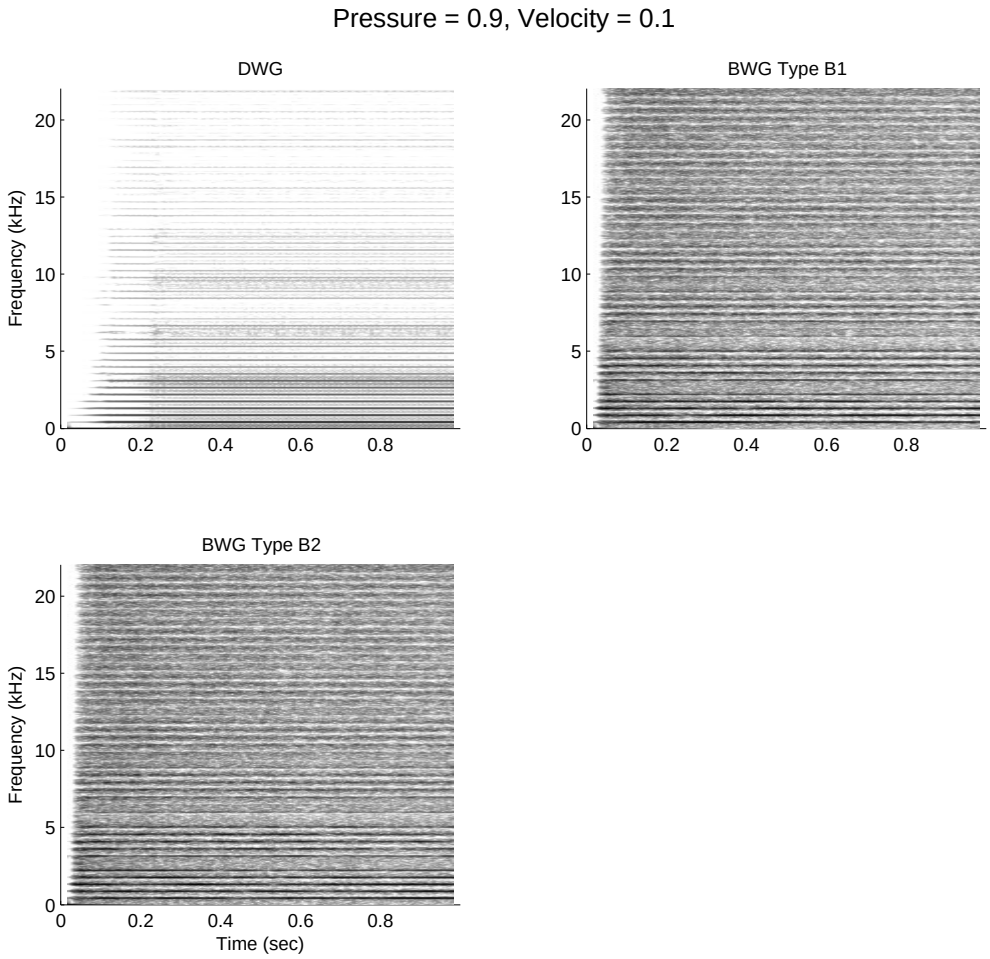


Figure C.68: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

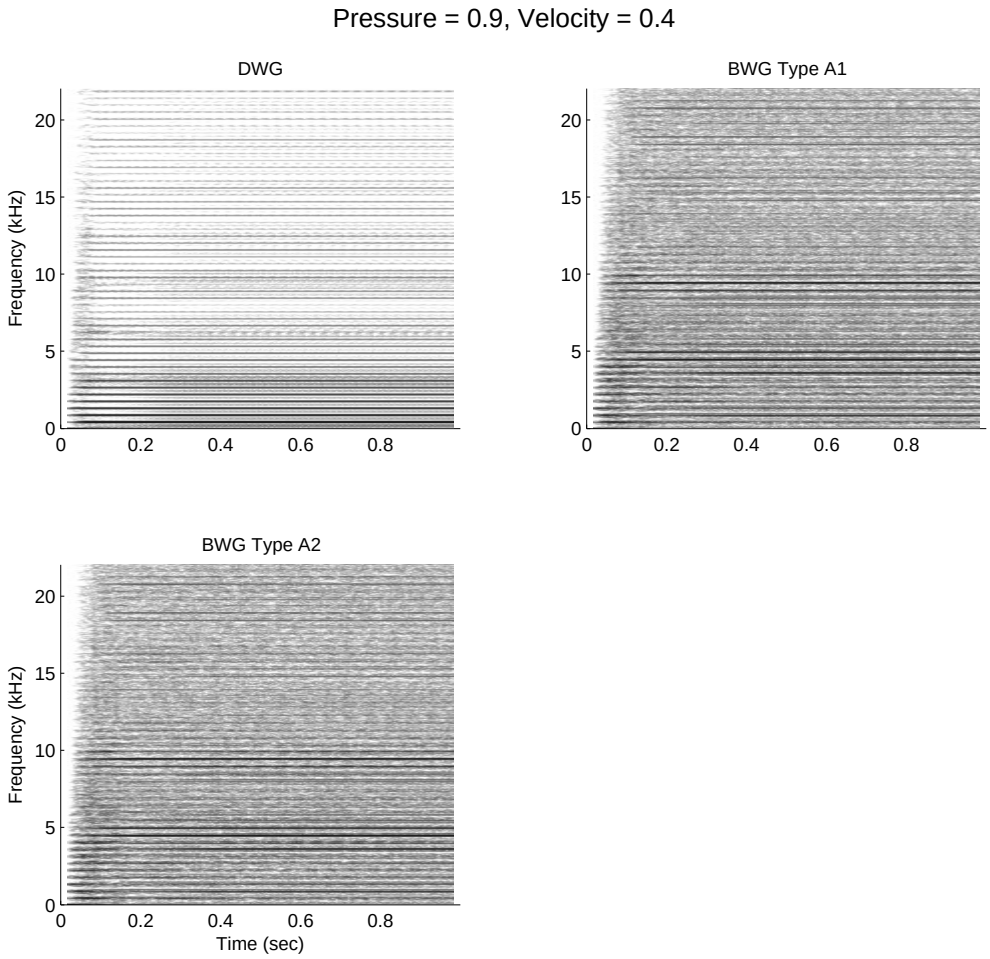


Figure C.69: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

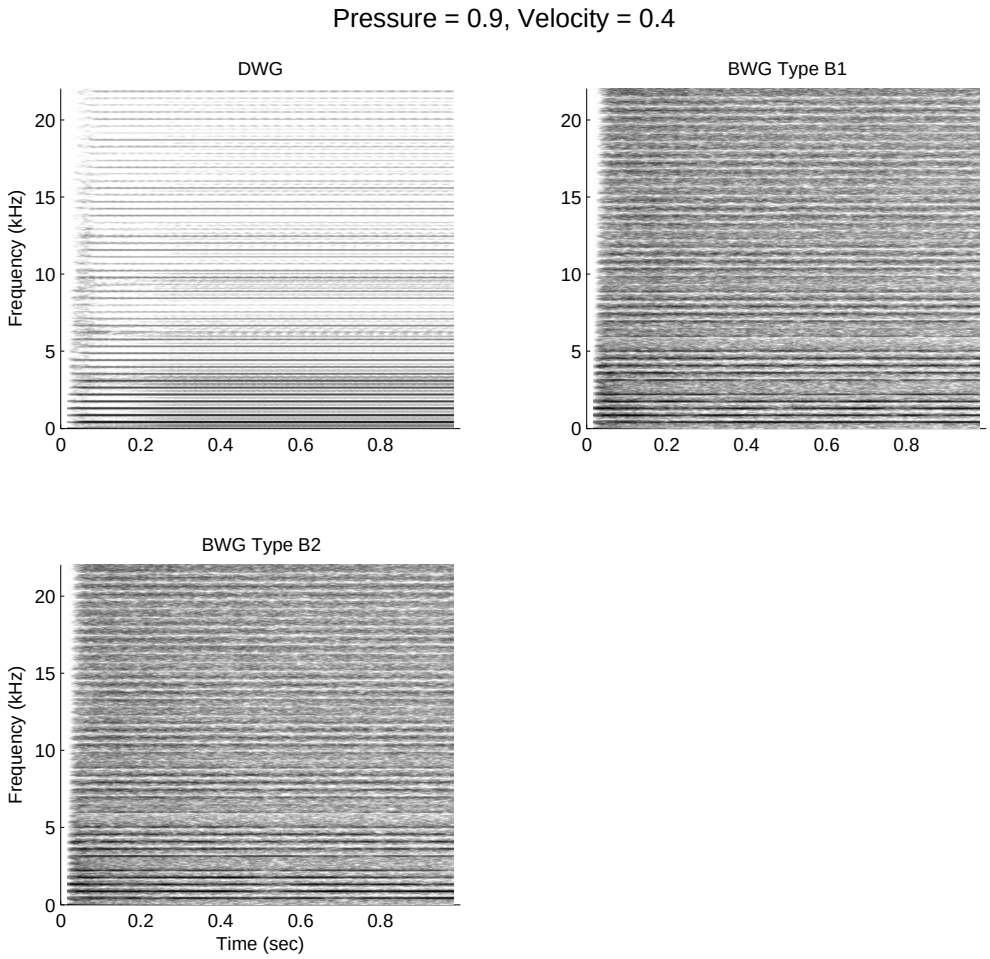


Figure C.70: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

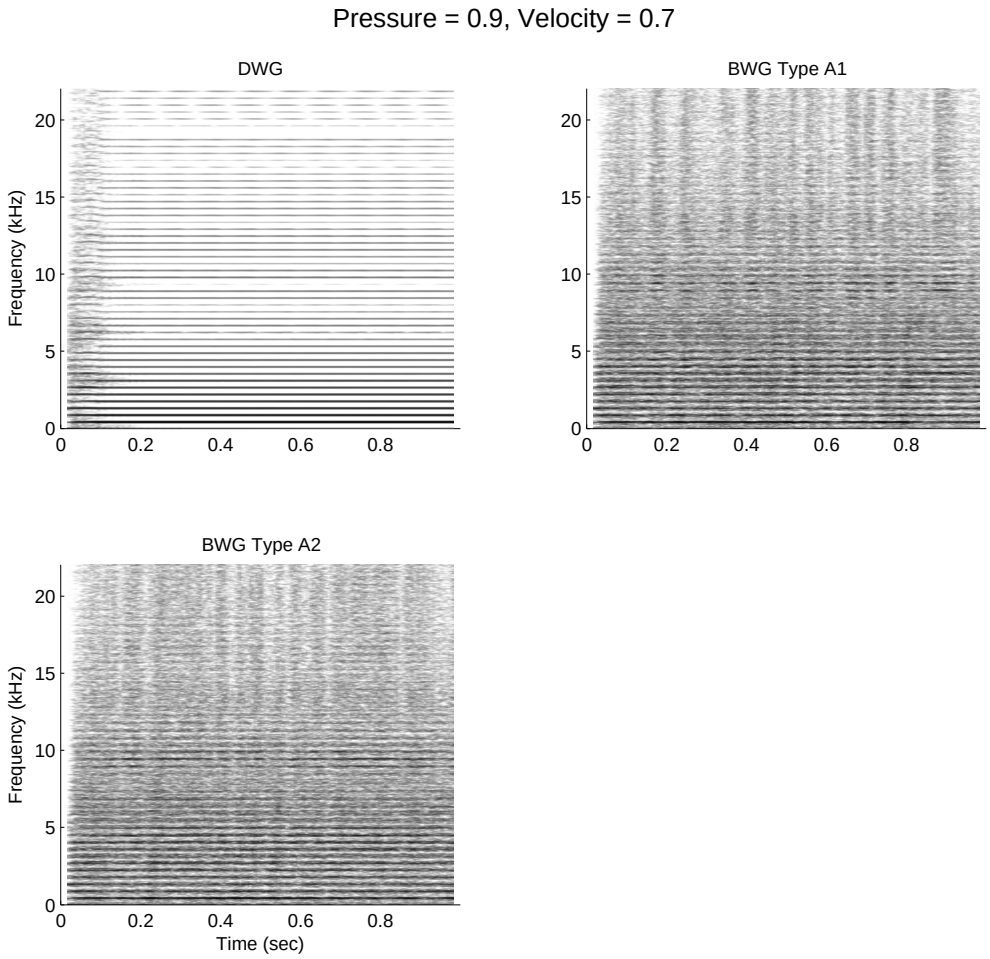


Figure C.71: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.

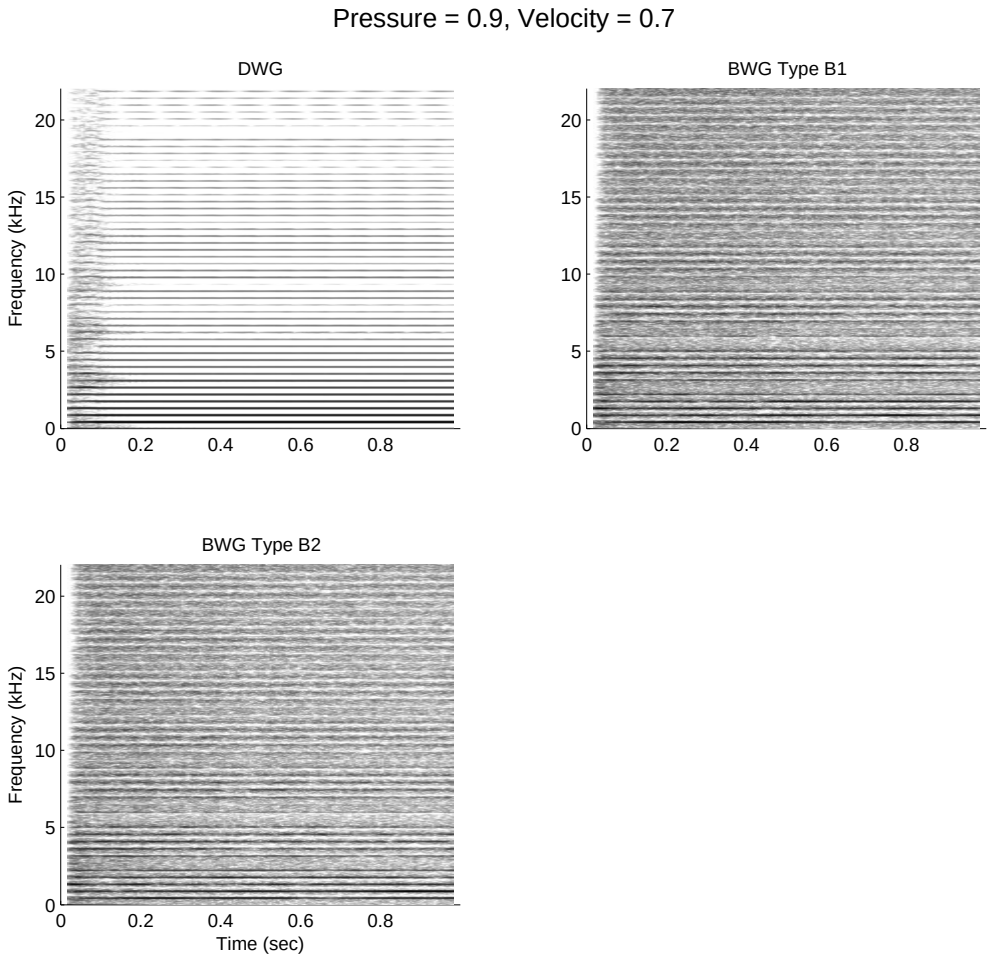


Figure C.72: Simulation of models of a bowed string with stiffness, tuned to 441 Hz. The spectrogram of the output of BWG models is shown, compared to that of a comparable DWG model.